MYS-8MMX Linux Software Evaluation Guide

File Status： [] Draft [√] Release	**FILE ID：**	MYIR-MYS-8MMX-SW-EG-EN-L5.4.3
	VERSION：	V1.0[DOC]
	AUTHOR：	Coin.Du
	CREATED：	2020-12-01
	UPDATED：	2021-01-15

Revision History

VERSION	AUTHOR	PARTICIPANT	DATE	DESCRIPTION
V1.0[DOC]	Coin.Du	Miller.Wu	2020122410	Initial Version:u-boot 2019，kernel 5.4.3，Yocto 3.0

CONTENT

1. Overview

The Linux software evaluation guide describes the testing steps and evaluation methods for core and peripheral resources running open source Linux systems on MYIR Electronics' development boards. This paper can be used as a preliminary evaluation guide or as a test guide for general system developers.

1.1. Hardware Resources

This document is applicable to MYIR’ s MYS-8MMX series boards, which are developed based on NXP company's high performance embedded ARM processor i.MX8MM series. For detailed configuration parameters for the hardware section, refer to MYS-8MMX Product Manual. At the same time, users will use some accessories during the evaluation test, see the list below.

Table 1-1.Optional Modules

Accessories	Interface	Description
Camera	CSI interface	MY-CAM003M（5MegaPixels） http://www.myir-tech.com/product/my_cam003m.htm
LVDS screen	LVDS interface	MY-LVDS070C（7 inches LVDS） http://www.myir-tech.com/product/my-lvds070c.htm
4G module	USB interface	EC20 https://www.quectel.com/cn/product/ec20r21minipcle.htm

1.2. Software Resources

The BSP of the MYS-8MMX series development board is based on the transplantation and modification of the official open source community version, and the system image is build by Yocto project. Bootloader,Kernel and all parts of the file system software resources are open in the form of source code,please check the “MYS-8MMX SDK Release Notes” for details.

The development board has been programmed with “myir-image-full” image when it leaves the factory, so you only need to power it up and use it.

1.3. Documents

Depending on the stages of using the development board, the SDK contains different categories of documentation and manuals in addition to release notes, evaluation guides, development guides, application notes, and frequently asked questions. Please refer to Table 2-3 of “MYS-8MMX SDK Release Notes” for the detailed list of documents.

1.4. Preparation

Before starting to evaluate the development board software, you need to do some necessary preparation and configuration of the development board environment, including proper hardware wiring, configuring debugging serial ports, setting up and other steps. For detailed steps, please refer to “MYS-8MMX Quick Start Guide”.

The following sections focus on how to evaluate and test the system's hardware resources and interfaces as well as software functions. Mainly with the help of some commonly used tools and commands under Linux, as well as our own examples of the application for testing. The software evaluation guide is divided into several parts, including: core components, peripheral interfaces, network applications, multimedia applications, development support applications, system tools and so on. The following chapters will make a comprehensive explanation for each part and describe in detail the specific evaluation methods and steps of each part of resources.

2. Core Components

In Linux, the PROC virtual file system is provided to query the parameters of various core resources and some common tools are provided to evaluate the performance of resources. The following will be specific to the CPU, memory, eMMC. The parameters of RTC and other core resources are also read and tested.

2.1. CPU

MYS-8MMX uses MIMX8MM6CVTKZAA/MIMX8MM6DVTLZAA as the main CPU,it is based on the high-performance 4-core Arm architecture ®-A53 64-bit RISC core, operating at Maximum commercial frequency 1.8 GHZ, industrial 1.6 GHZ. Each CPU core of the Cortex-A53 processor includes a 32 KByte L1 instruction cache, a 32 KByte L1 data cache, and a 512 KByte L2 cache. The MIMX8MM6CVTKZAA/MIMX8MM6DVTLZA device is also embedded with the Cortex®-M4 32-bit RISC core and can operate at up to 400 MHz. The core functionality of Cortex-M4 is floating point unit (FPU) single precision, which supports Arm® single precision data processing instructions and data types. Cortex-m4 supports a complete set of DSP instructions and a memory protection unit (MPU) to enhance application security.

1) View the CPU Information

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

```
root@mys-8mmx:~# cat /proc/cpuinfo
processor       : 0
BogoMIPS        : 16.00
Features        : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant     : 0x0
CPU part        : 0xd03
CPU revision      : 4

processor         : 1
BogoMIPS          : 16.00
Features          : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant       : 0x0
CPU part          : 0xd03
CPU revision      : 4

processor         : 2
BogoMIPS          : 16.00
Features          : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant       : 0x0
CPU part          : 0xd03
CPU revision      : 4

processor         : 3
BogoMIPS          : 16.00
Features          : fp asimd evtstrm aes pmull sha1 sha2 crc32 cpuid
CPU implementer : 0x41
CPU architecture: 8
CPU variant       : 0x0
CPU part          : 0xd03
CPU revision      : 4
```

- processor：The number of logical processing cores in a system can be either a physical core for multi-core processors or a virtual logical core using hyperthreading technology
- model name ： The name and number of the CPU belongs to

- bogomips：A rough measure of the number of millions of instructions the CPU runs per second at kernel startup（Million Instructions Per Second）

2) View CPU Utilization

```
root@mys-8mmx:~# top
top - 08:00:32 up 13 min,  2 users,  load average: 1.00, 0.95, 0.61
Tasks: 125 total,   2 running, 123 sleeping,   0 stopped,   0 zombie
%Cpu(s): 24.6 us,  0.3 sy,  0.0 ni, 74.6 id,  0.0 wa,  0.4 hi,  0.1 si,  0.0 st
MiB Mem :   1943.6 total,    601.7 free,   1259.4 used,     82.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    594.0 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME
+ COMMAND
    700 root      20   0 1050476   1.0g   1056 R  98.2  52.7  12:34.70 me
mtester
    600 avahi     20   0    4848   2388   2140 S   0.3   0.1   0:00.57 avahi
-daemon
    686 root      20   0    2868   1996   1716 S   0.3   0.1   0:07.12 drop
bear
    703 root      20   0    3424   2212   1764 R   0.3   0.1   0:00.05 top

      1 root      20   0   10188   6888   4984 S   0.0   0.3   0:03.26 syste
md
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.01 kthrea
dd
```

- %us：Represents the CPU utilization of the user space program (not scheduled by NICE)
- %sy：Represents the CPU utilization of system space, mainly kernel programs
- %ni：Represents the CPU usage of a program that has been scheduled through NICE in user space
- %id：idle CPU

3) Gets CPU Temperature

There is a TMU module inside the CPU chip to detect the CPU temperature, and the temperature range is 10~125.

```
root@mys-8mmx:~# cat /sys/class/thermal/thermal_zone0/temp
49000
```

The number shown above is in the unit of one thousandth of degree centigrade, the result divided by 1000 is the current temperature.

4) CPU Stress Test

There are many ways to test the CPU pressure. For example, “bc” command can be used to calculate the PI, and we can test the stability of the CPU in the process of operation by this means.

```
root@mys-8mmx:~# echo "scale=5000; 4*a(1)" | bc -l -q &
[1] 707
```

The above command will calculate the PI in the background and accurate to 5000 decimal places. The calculation process needs a period of time. At this time, we can check the variation of CPU utilization by the top command, as follows：

```
root@mys-8mmx:~# top
top - 08:02:18 up 15 min,  2 users,  load average: 1.44, 1.08, 0.69
Tasks: 128 total,   3 running, 125 sleeping,   0 stopped,   0 zombie
%Cpu(s): 49.7 us,  0.2 sy,  0.0 ni, 49.4 id,  0.0 wa,  0.5 hi,  0.2 si,  0.0 st
MiB Mem :   1943.6 total,    600.1 free,   1260.7 used,     82.8 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.    592.6 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME
+ COMMAND
    700 root      20   0 1050476   1.0g   1056 R  99.0  52.7  14:19.43 me
mtester
    707 root      20   0    2728   1704   1452 R  98.7   0.1   0:33.14 bc
    711  root      20    0    3428    2136    1680  R    1.3    0.1    0:00.11  top

    686  root      20    0    2868    1996    1716  S    0.3    0.1    0:07.39  drop
bear
.........
```

After about 1 minutes, the result of PI was calculated. The CPU usage was very high and there were no exceptions, indicating that the CPU stress test had passed.By increasing the accuracy requirement, the test pressure can be further improved.

```
root@mys-8mmx:~#  3.14159265358979323846264338327950288419716939937
510582097494459230781640628620899862803482534211706798214808651328230664709384460955058223172535940812848111745028410270193852110555964462294895493038196442881097566593344612847564823378678316527120190914564856692346034861045432664821339360726024914127372458700660631558817488152092096282925409171536436789259036001133053054882046652138414695194151160943305727036575959195309218611738193261179310511854807446237996274956735188575272489122793818301194912983367336244065664308602139494639522473719070217986094370277053921717629317675238467481846766940513200056812714526356082778577134275778960917363717872146844090122495343014654958537105079227968925892354201995611212902196086403441815981362977477130996051870721134999999837297804995105
................
```

5) CPU operating frequency

CPU support dynamic frequency modulation, but also support fixed frequency, and synchronous operation of 4 Cortex A53 cores.

- **Check CPU frequency**

```
root@mys-8mmx:~#  cat  /sys/bus/cpu/devices/cpu0/cpufreq/cpuinfo_cur_freq
1200000
```

- **Check CPU maximum operating frequency**

```
root@mys-8mmx:~#  cat  /sys/bus/cpu/devices/cpu0/cpufreq/cpuinfo_max_freq
1800000
```

- **Check out all CPU working patterns**

```
root@mys-8mmx:~# cat /sys/bus/cpu/devices/cpu0/cpufreq/scaling_available_governors
userspace ondemand performance schedutil
```

Description of several working modes of the CPU:

- userspace：User mode, in any case will control the CPU running within the frequency range of the configuration, the configuration of the user's own added power saving settings. in this scenario mode, reducing the maximum operating frequency of the CPU can prolong the battery standby time, but it will also reduce the wake-up speed of the machine. it is recommended that this option should not be used
- ondemand:Refers to the usual low-speed operation, when the system load increases automatically increase the frequency. Running in this mode will not reduce the performance due to frequency reduction, but also save electricity and reduce temperature.
- performance:High performance mode, running at the highest frequency of your set range, even if the system load is very low, the frequency of cpu is the highest. Good performance because the CPU itself does not need resources to adjust the frequency, but the electricity consumption is faster and the temperature is higher.
- Schedutil:Adjust the CPU frequency based on the scheduler.

- **Check CPU current working mode**

```
root@mys-8mmx:~# cat /sys/bus/cpu/devices/cpu0/cpufreq/scaling_governor
ondemand
```

This allows CPU to be at the highest frequency by setting the CPU performance mode.

```
root@mys-8mmx:~# echo "performance" > /sys/bus/cpu/devices/cpu0/cpufreq/scaling_governor
root@mys-8mmx:~# cat /sys/bus/cpu/devices/cpu0/cpufreq/cpuinfo_cur_freq
1800000
```

2.2. M4

MYS-8MMX is equipped with a heterogeneous Cortex-M4 coprocessor. You can run Linux and RTOS at the same time .this section mainly introduces the use M4 coprocessors.

m4 may involve sharing resources with A53 cores at run time. Here are the conflicting resources，for example：ECSPI0/ECSPI2, GPIO1/GPIO5, GPT1, I2C3, I2S3, WDOG1, UART4, PWM3, SDMA1，So when you use it together, you need to turn off these resources on the A53 side, and you need to use myb-imx8mm-rpmsg-m4.dtb device tree here. The M4 side uses the uart4 as the serial port.This section describes how to use M4, and how to compile M4 programs.

1) How to Use Cortex-M4

There are two serial ports, one on the development board debug serial port (J25), the other on the M4 side of the uart4 serial port (J22),and then set the parameters of two serial ports.

- **View partition information**

  Enters the uboot mode to view files that exist in the vfat partition:

```
u-boot=>
u-boot=> mmc list
FSL_SDHC: 1 (SD)
FSL_SDHC: 2
u-boot=> fatls mmc 1
 35238400   Image
     6000   imx8mm_m4_TCM_hello_world.bin
    16528   imx8mm_m4_TCM_rpmsg_lite_pingpong_rtos_linux_remote.bin
    16028   imx8mm_m4_TCM_rpmsg_lite_str_echo_rtos.bin
    40092   imx8mm_m4_TCM_sai_low_power_audio.bin
    40649   mys-imx8mm-evk.dtb
    43376   mys-imx8mm-lt8912-atk-10-1.dtb
    43376   mys-imx8mm-lt8912-hontron-7.dtb
   373608   tee.bin
```

```
                System Volume Information/
     41051    mys-imx8mm-evk-rpmsg.dtb

11 file(s), 1 dir(s)

u-boot=>
```

- imx8mm_m4_TCM_hello_world.bin : m4 program
- imx8mm_m4_TCM_rpmsg_lite_pingpong_rtos_linux_remote.bin : m4program
- imx8mm_m4_TCM_rpmsg_lite_str_echo_rtos.bin : m4 program
- imx8mm_m4_TCM_sai_low_power_audio.bin : m4 program
- mys-imx8mm-evk-rpmsg.dtb : m4 device tree

- **Set M4 Device Tree**

kernel loaded device tree is determined by the fdt_file variable, set to m4 dedicated device tree:

```
u-boot=> print fdt_file
fdt_file=mys-imx8mm-lt8912-hontron-7.dtb
u-boot=> setenv fdt_file mys-imx8mm-evk-rpmsg.dtb
u-boot=> save
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=> printenv fdt_file
fdt_file=mys-imx8mm-evk-rpmsg.dtb
u-boot=>
```

- **Set m4 start parameter**

m4 boot process also loads m4 program into memory and then reboot:

```
u-boot=> setenv m4_image imx8mm_m4_TCM_rpmsg_lite_str_echo_rtos.bin
u-boot=> setenv m4_boot_temp_addr 0x48000000
u-boot=> setenv m4_boot_addr 0x7E0000
u-boot=> setenv m4_run 'fatload mmc ${mmcdev}:${mmcpart} ${m4_boot_temp_addr} ${m4_image};cp.b ${m4_boot_temp_addr} ${m4_boot_addr} 0x20000; bootaux ${m4_boot_addr}'
u-boot=> setenv mmcboot "run m4_run;${mmcboot}"
u-boot=> save
```

```
Saving Environment to MMC... Writing to MMC(1)... OK
u-boot=>
```

- **Test m4 program**

The m4 startup parameters have been set up at this time, only need to restart the development board, then while starting the system, will also start the program in the m4. Then execute the sending data command on the A53 side of the serial port, you can see M4 side of the serial port corresponding printing.

A53 Serial:

```
root@mys-8mmx:~# modprobe imx_rpmsg_tty
[   50.488862] imx_rpmsg_tty virtio0.rpmsg-virtual-tty-channel-1.-1.30: new cha
nnel: 0x400 -> 0x1e!
[   50.498510] Install rpmsg tty driver!
root@mys-8mmx:~# echo "hi m4!" > /dev/ttyRPMSG30
root@mys-8mmx:~# echo "hi m4!" > /dev/ttyRPMSG30
root@mys-8mmx:~# echo "hello world" > /dev/ttyRPMSG30
```

M4 Serial:

```
Get Message From Master Side : "hi m4!" [len : 6]
Get New Line From Master Side
Get Message From Master Side : "hi m4!" [len : 6]
Get New Line From Master Side
Get Message From Master Side : "hello world" [len : 11]
Get New Line From Master Side
```

2) M4 demo compilation

You can obtain M4 source code and M4 tool chain from the official website, or you can download the following two files directly from the MYIR path:

- m4source SDK_2.8.0_EVK-MIMX8MM.tar.gz
- m4 tool chain gcc-arm-none-eabi-7-2017-q4-major-linux.tar.bz2

Demo compiled

```
myir@myir-server1:~/MYS-8MMX-M4/m4$ ls -l
Total 102348
```

```
-rwxr--r-- 1 myir myir 99857645 12月 16 09:44 gcc-arm-none-eabi-7-2017-q4-major-linux.tar.bz2
-rwxr--r-- 1 myir myir  4941322 12月 16 09:44 SDK_2.8.0_EVK-MIMX8MM.tar.gz
```

- **Decompressing files**

Extract M4 source and compile chains:

```
myir@myir-server1:~/MYS-8MMX-M4/m4$ mkdir toolchains
myir@myir-server1:~/MYS-8MMX-M4/m4$ tar -jxf gcc-arm-none-eabi-7-2017-q4-major-linux.tar.bz2 -C toolchains/
myir@myir-server1:~/MYS-8MMX-M4/m4$ mkdir SDK_2.8.0_EVK-MIMX8MM
myir@myir-server1:~/MYS-8MMX-M4/m4$ tar -zxf SDK_2.8.0_EVK-MIMX8MM.tar.gz -C SDK_2.8.0_EVK-MIMX8MM/
```

- **Load compile chain**

Sets the value of the ARMGCC_DIR environment variable：

```
myir@myir-server1:~/MYS-8MMX-M4/m4$ export ARMGCC_DIR=`pwd`/toolchains/gcc-arm-none-eabi-7-2017-q4-major
myir@myir-server1:~/MYS-8MMX-M4/m4$ env | grep ARMGCC
ARMGCC_DIR=/home/myir/MYS-8MMX-M4/m4/toolchains/gcc-arm-none-eabi-7-2017-q4-major
myir@myir-server1:~/MYS-8MMX-M4/m4$
```

- **M4 demo compiled**

Go to the compilation directory, then execute the clean.sh,build_debug.sh script in turn, you can generate the execution file.

```
myir@myir-server1:~/MYS-8MMX-M4/m4$ cd SDK_2.8.0_EVK-MIMX8MM/boards/evkmimx8mm/demo_apps/hello_world/armgcc/
myir@myir-server1:~/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/boards/evkmimx8mm/demo_apps/hello_world/armgcc$ ./clean.sh
myir@myir-server1:~/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/boards/evkmimx8mm/demo_apps/hello_world/armgcc$ ./build_debug.sh
-- TOOLCHAIN_DIR: /home/myir/MYS-8MMX-M4/m4/toolchains/gcc-arm-none-eabi-7-2017-q4-major
```

```
CMake Deprecation Warning at /usr/local/share/cmake-3.8/Modules/CMakeForceCompiler.cmake:69 (message):
  The CMAKE_FORCE_C_COMPILER macro is deprecated.  Instead just set
  CMAKE_C_COMPILER and allow CMake to identify the compiler.
Call Stack (most recent call first):
  /home/myir/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/tools/cmake_toolchain_files/armgcc.cmake:33 (CMAKE_FORCE_C_COMPILER)
  /usr/local/share/cmake-3.8/Modules/CMakeDetermineSystem.cmake:85 (include)
  CMakeLists.txt


CMake Deprecation Warning at /usr/local/share/cmake-3.8/Modules/CMakeForceCompiler.cmake:83 (message):
  The CMAKE_FORCE_CXX_COMPILER macro is deprecated.  Instead just set
  CMAKE_CXX_COMPILER and allow CMake to identify the compiler.
Call Stack (most recent call first):
  /home/myir/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/tools/cmake_toolchain_files/armgcc.cmake:34 (CMAKE_FORCE_CXX_COMPILER)
  /usr/local/share/cmake-3.8/Modules/CMakeDetermineSystem.cmake:85 (include)
  CMakeLists.txt


-- BUILD_TYPE: debug
-- TOOLCHAIN_DIR: /home/myir/MYS-8MMX-M4/m4/toolchains/gcc-arm-none-eabi-7-2017-q4-major
CMake Deprecation Warning at /usr/local/share/cmake-3.8/Modules/CMakeForceCompiler.cmake:69 (message):
  The CMAKE_FORCE_C_COMPILER macro is deprecated.  Instead just set
  CMAKE_C_COMPILER and allow CMake to identify the compiler.
Call Stack (most recent call first):
  /home/myir/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/tools/cmake_toolchain_files/armgcc.cmake:33 (CMAKE_FORCE_C_COMPILER)
  CMakeFiles/3.8.2/CMakeSystem.cmake:6 (include)
  CMakeLists.txt


CMake Deprecation Warning at /usr/local/share/cmake-3.8/Modules/CMakeForceCompiler.cmake:83 (message):
  The CMAKE_FORCE_CXX_COMPILER macro is deprecated.  Instead just set
  CMAKE_CXX_COMPILER and allow CMake to identify the compiler.
Call Stack (most recent call first):
  /home/myir/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/tools/cmake_toolchain_files/armgcc.cmake:34 (CMAKE_FORCE_CXX_COMPILER)
  CMakeFiles/3.8.2/CMakeSystem.cmake:6 (include)
  CMakeLists.txt


-- BUILD_TYPE: debug
-- The ASM compiler identification is GNU
-- Found assembler: /home/myir/MYS-8MMX-M4/m4/toolchains/gcc-arm-none-eabi-7-2017-q4-major/bin/arm-none-eabi-gcc
-- Configuring done
-- Generating done
-- Build files have been written to: /home/myir/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/boards/evkmimx8mm/demo_apps/hello_world/armgcc
Scanning dependencies of target hello_world.elf
[  5%] Building C object CMakeFiles/hello_world.elf.dir/home/myir/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/boards/evkmimx8mm/demo_apps/hello_world/clock_config.c.obj
[ 10%] Building C object CMakeFiles/hello_world.elf.dir/home/myir/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/devices/MIMX8MM6/drivers/fsl_rdc.c.obj
[ 15%] Building C object CMakeFiles/hello_world.elf.dir/home/myir/MYD-C8MM
....................................
```

```
[ 45%] Building ASM object CMakeFiles/hello_world.elf.dir/home/myir/MYS-8M
MX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/devices/MIMX8MM6/gcc/startup_MIMX8
MM6_cm4.S.obj
[ 45%] Building C object CMakeFiles/hello_world.elf.dir/home/myir/MYD-C8MM
[ 85%] Building C object CMakeFiles/hello_world.elf.dir/home/myir/MYS-8MMX
-M4/m4/SDK_2.8.0_EVK-MIMX8MM/devices/MIMX8MM6/drivers/fsl_clock.c.obj
[ 90%] Building C object CMakeFiles/hello_world.elf.dir/home/myir/MYS-8MMX
-M4/m4/SDK_2.8.0_EVK-MIMX8MM/devices/MIMX8MM6/utilities/str/fsl_str.c.obj
[ 95%] Building C object CMakeFiles/hello_world.elf.dir/home/myir/MYS-8MMX
-M4/m4/SDK_2.8.0_EVK-MIMX8MM/components/serial_manager/serial_port_uart.
c.obj
[100%] Linking C executable debug/hello_world.elf
[100%] Built target hello_world.elf
myir@myir-server1:~/MYS-8MMX-M4/m4/SDK_2.8.0_EVK-MIMX8MM/boards/evk
mimx8mm/demo_apps/hello_world/armgcc$
```

2.3. Graphic

MYS-8MMX chip has a GC NanoUltra GPU module inside, supporting 2 D,3D acceleration, supporting OpenGL ES1.1,OpenGL ES2.0,OpenVG 1.1.

The specific operation process refers to the subsequent 6.1 graphics and image processing chapter.

2.4. Memory

MYS-8MMX memory default is 2 G.

1) Check the Memory Information

The parameter information of memory in the system can be obtained by reading the "*/proc/meminfo*" file.

```
root@mys-8mmx:~# cat /proc/meminfo
MemTotal:          1990276 kB
MemFree:            1667740 kB
MemAvailable:      1660060 kB
Buffers:              6572 kB
Cached:               60140 kB
SwapCached:                0 kB
Active:             63260 kB
Inactive:          42768 kB
Active(anon):       40096 kB
Inactive(anon):       9388 kB
Active(file):     23164 kB
Inactive(file):   33380 kB
Unevictable:            0 kB
Mlocked:                  0 kB
SwapTotal:               0 kB
SwapFree:                 0 kB
Dirty:                   4 kB
Writeback:               0 kB
AnonPages:           39340 kB
Mapped:              25900 kB
Shmem:                10144 kB
KReclaimable:       18092 kB
Slab:               52356 kB
SReclaimable:       18092 kB
SUnreclaim:         34264 kB
```

```
KernelStack:          2352  kB
PageTables:            1308  kB
NFS_Unstable:             0  kB
Bounce:                   0  kB
WritebackTmp:              0  kB
CommitLimit:        995136  kB
Committed_AS:         87136  kB
VmallocTotal:    135290159040  kB
VmallocUsed:          8932  kB
VmallocChunk:             0  kB
Percpu:               1296  kB
HardwareCorrupted:        0  kB
AnonHugePages:             0  kB
ShmemHugePages:             0  kB
ShmemPmdMapped:              0  kB
FileHugePages:            0  kB
FilePmdMapped:              0  kB
CmaTotal:           655360  kB
CmaFree:            516532  kB
HugePages_Total:          0
HugePages_Free:           0
HugePages_Rsvd:            0
HugePages_Surp:           0
Hugepagesize:         2048  kB
Hugetlb:                  0  kB
```

Partial parameter description:

- MemTotal : All available RAM sizes, physical memory minus reserved bits and kernel usage
- MemFree : Current amount of free memory
- Buffers : Size used to cache block devices
- Cached : 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 has not been used much recently

2) Get memory usage

The "free" command can be used to read memory usage, with the "-m" parameter representing Mega Byte.

```
root@mys-8mmx:~# free -m
              total        used        free      shared  buff/cache   available
Mem:           1943         232        1628           9          82        1620
Swap:             0           0           0
```

- total : Total Memory
- used : The amount of memory used
- free : The amount of memory available

3) Memory Stress Test

The "memtester" tool under Linux system can be used to test the pressure of the existing memory of the system by giving the size and times of test memory. If the memory test is specified once and the memory under test is 1GByte, the test method is as follows.

```
root@mys-8mmx:~# memtester 1024M 1
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 1024MB (1073741824 bytes)
got  1024MB (1073741824 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

Done.
```

2.5. e•MMC

eMMC is a communication and mass data storage device, including a Multi-Media Card (MMC) interface, an NAND Flash component, and a controller on an advanced signal bus, which is compliant with the MMC system specification. Its cost, small size, Flash technology independence, and high data throughput make it an ideal choice for embedded applications.The commercial development board equipped with 8Gb Sandisk eMMC,while the industrial development board equipped with 8Gb Micron e.MMC.Commercial eMMC reading and writing speed is basically twice the industrial level. The following is an example of industrial grade.

1) Check eMMC Capacity

The eMMC partition information and capacity can be queried through the "fdisk -l" command as follows.

```
root@mys-8mmx:~# fdisk -l
Disk /dev/mtdblock0: 32 MiB, 33554432 bytes, 65536 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
Disk /dev/mmcblk2: 7.1 GiB, 7616856064 bytes, 14876672 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: dos
Disk identifier: 0x6e9251d4

Device         Boot  Start     End Sectors  Size Id Type
/dev/mmcblk2p1 *     16384  186775  170392 83.2M  c W95 FAT32 (LBA)
/dev/mmcblk2p2      196608 7739611 7543004  3.6G 83 Linux
```

- /dev/mmcblk2p1 : Storage of kernel and dtb files
- /dev/mmcblk2p2 : Used to store the root file system

This / dev/mmcblk2p1 starts at 16384 blocks, with bootloader and partition table information saved in front.

2) View the eMMC Partition Information

By “df” command, you can check eMMC partition information, usage, mount directory and other information.

```
root@mys-8mmx:~# df -h
Filesystem          Size   Used  Avail  Use%  Mounted  on
/dev/root            3.5G   2.1G   1.2G   64%  /
devtmpfs             649M   4.0K   649M    1%  /dev
tmpfs                972M      0   972M    0%  /dev/shm
tmpfs                972M   9.0M   963M    1%  /run
tmpfs                972M      0   972M    0%  /sys/fs/cgroup
tmpfs                972M   4.0K   972M    1%  /tmp
tmpfs                972M   196K   972M    1%  /var/volatile
tmpfs                195M   4.0K   195M    1%  /run/user/0
/dev/mmcblk2p1        84M    35M    49M   42%  /run/media/mmcblk2p1
```

- ➢ /dev/root : Root file system, mounted to the root directory
- ➢ tmpfs : Memory virtual file system, mounted to a different directory
- ➢devtmpfs : Used to create dev for the system
- ➢ /dev/mmcblk2p1 : Used to store kernel and dtb files, mounted in / run/media/mmcblk2p1 directory

3) eMMC Performance Test

The performance test mainly tests the reading and writing speed of files by eMMC in Linux system. The test is generally combined with the dual command of “time” and “dd” .

- **Write Performance**

```
root@mys-8mmx:~# time dd if=/dev/zero of=tempfile bs=1M count=100 con
v=fdatasync
100+0 records in
```

```
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2.94129 s, 35.7 MB/s

real    0m2.946s
user    0m0.000s
sys     0m0.495s
```

When you use the dd command to write a file, you need to add the “conv = fdatasync” parameter,which means that after using the dd command to write multiple times, it will be synchronized to the disk(flush-cached). Because the disk is usually written to the cache first and then returned before it is written to the disk. Here we test that the write disk speed is 35.7MB/s.

- **Read Performance**

In embedded system, it is often necessary to test the system's performance of reading. If you want to test reading files directly from disk, you need to ignore the impact of cache. In this case, you can specify the parameter “iflag = direct, Nonblock ”.

```
root@mys-8mmx:~#  time dd if=tempfile of=/dev/null bs=1M count=100 ifla
g=direct,nonblock
100+0 records in
100+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 0.396706 s, 264 MB/s

real    0m0.401s
user    0m0.007s
sys     0m0.016s
```

The test results show that reading speed without cache is 264MB/s.

2.6. QSPI

MYS-8MMX equipped with a 32 M qspi, it can be used to save data. Also, the memory write area must be the erased area; the erase is by block (0x10000) as the minimum unit.

1) Read and write under the File System

Here are the test steps under the file system and some commands qspi read and write:

- hexdump :View partition content
- mtd_debug read : Read qspi data to file
- mtd_debug wirte：Writes file data to qspi
- mtd_debug erase ：Erase qspi data

- **View stored content**

Use hexdump to view QSPI stored content:

```
root@mys-8mmx:~# hexdump -C /dev/mtd0
00000000  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
*
02000000
```

- **Write Data**

Write data to the QSPI via the mtd_debug wirte command and view the write content:

```
root@mys-8mmx:~# echo "myir qspi test" > qspi_test.txt
root@mys-8mmx:~# mtd_debug write /dev/mtd0 0x10 14 qspi_test.txt
Copied 14 bytes from qspi_test.txt to address 0x00000010 in flash
root@mys-8mmx:~# hexdump -C /dev/mtd0
00000000  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
00000010  6d 79 69 72 20 71 73 70  69 20 74 65 73 74 ff ff  |myir qspi test..|
00000020  ff ff ff ff ff ff ff ff  ff ff ff ff ff ff ff ff  |................|
*
```

```
02000000
```

- **Read Data**

Read QSPI data by mtd_debug read command:

```
root@mys-8mmx:~# mtd_debug read /dev/mtd0 0x10 14 read.txt
Copied 14 bytes from address 0x00000010 in flash to read.txt
root@mys-8mmx:~# cat read.txt
myir qspi testroot@mys-8mmx:~#
```

2) Read and write under qspi uboot

The following are the steps qspi reading in the uboot phase. The memory operation instructions are generally mw and md.

- **Uboot Memory Read Command**

  - **mw command**

Execute memory write command ‘mw’ under uboot, will prompt mw usage:

```
u-boot=> mw
mw - memory write (fill)
Usage:
mw [.b, .w, .l, .q] address value [count]
```

b：8 bit w：16 bit l：32 bit (default value), representing the number of bits written each time; The ‘address’ represents the address to write to memory, the ‘value’ represents the value to write, and the ‘count’ represents the number to write from the all hexadecimal numbers.

  - **md command**

Execute memory read command’ md’ under uboot,will prompt mw usage:

```
u-boot=> md
md - memory display
Usage:
md [.b, .w, .l, .q] address [# of objects]
u-boot=>
```

b/w/l means the same as above, the default is 32-bit output.

- **Initialization QSPI**

The sf command of the SPI flash subsystem is needed here, Execute memory read command’ sf’ under uboot,will prompt mw usage.

Start the development board, press any key to enter the uboot shell interface, execute the following command to initialize the qspi：

```
u-boot=>
u-boot=> sf probe
SF: Detected n25q256ax1 with page size 256 Bytes, erase size 4 KiB, total 32
MiB
```

- **Read Data after Erasing Memory**

After erasing some memory, read qspi contents:

```
u-boot=> mw.b 0x48000000 0xff 0x10
u-boot=> md 0x48000000 0x10
48000000: ffffffff ffffffff ffffffff ffffffff    ................
48000010: 00000000 00000000 00002014 00000000    ......... ......
48000020: 00000000 00000000 00000000 00000000    ................
48000030: 00002011 00000000 00000001 00000000    . ..............
u-boot=> sf read 0x48000000 0x10 0x10
device 0 offset 0x10, size 0x10
SF: 16 bytes @ 0x10 Read: OK
u-boot=> md 0x48000000 0x10
u-boot=> md 0x48000000 0x10
48000000: 7269796d 70737120 65742069 ffff7473    myir qspi test..
48000010: 00000000 00000000 00002014 00000000    ......... ......
48000020: 00000000 00000000 00000000 00000000    ................
48000030: 00002011 00000000 00000001 00000000    . ..............
```

Read 16 bytes from QSPI offset address 0 x10 to memory 0x48000000 address with the sf read command. Since data has previously been written at QSPI 0 x10 address, you can see the "myir qspi test" content written in the file system.

- **Write Data after Erasing Memory**

Write 16 bytes of data back to memory after erasing part of memory, then write 16 bytes of 0x12 to qspi 0x100 offset：

```
u-boot=> mw.b 0x48000000 0xff 0x10
u-boot=> mw.b 0x48000000 0x12 0x10
u-boot=> md 0x48000000 0x10
48000000: 12121212 12121212 12121212 12121212    ................
48000010: 00000000 00000000 00002014 00000000    ......... ......
48000020: 00000000 00000000 00000000 00000000    ................
48000030: 00002011 00000000 00000001 00000000    . ..............
u-boot=> sf write 0x48000000 0x100 0x10
device 0 offset 0x100, size 0x10
SF: 16 bytes @ 0x100 Written: OK
u-boot=>
```

- **Read again after Erasing Memory**

After erasing some memory contents, read the contents written to the qspi again into memory:

```
u-boot=> mw.b 0x48000000 0xff 0x10
u-boot=> sf read 0x48000000 0x100 0x10
device 0 offset 0x100, size 0x10
SF: 16 bytes @ 0x100 Written: OK
u-boot=> mw.b 0x48000000 0xff 0x10
u-boot=> sf read 0x48000000 0x100 0x10
device 0 offset 0x100, size 0x10
SF: 16 bytes @ 0x100 Read: OK
u-boot=> md 0x48000000 0x10
48000000: 12121212 12121212 12121212 12121212    ................
48000010: 00000000 00000000 00002014 00000000    ......... ......
48000020: 00000000 00000000 00000000 00000000    ................
48000030: 00002011 00000000 00000001 00000000    . ..............
```

You can read the written data from the QSPI normally, and you can also refer to the file system to read the qspi content.

2.7. RTC

MYS-8MMX has internal RTC (snvs) and external RTC (RX8025). The function of RTX is to synchronize real time after system startup. If you're using the inside RTC, then when you press the ON/OFF button of the development board instead of unplugging the power cord to turn off the machine, the time is still in sync after rebooting.

1) View RTC Devices in Linux System

```
root@mys-8mmx:~# ls -l /dev/rtc*
lrwxrwxrwx 1 root root         4 Nov 18 09:47 /dev/rtc -> rtc0
crw------- 1 root root 251, 0 Nov 18 09:47 /dev/rtc0
```

“/dev/rtc0” is the default RTC device driver node for applications.

```
root@mys-8mmx:~# cat /sys/class/rtc/rtc0/name
rtc-rx8025 1-0032
```

2) Set System Time

Set the system time to Wed Dec 16 11:30:10 UTC 2020：

```
root@mys-8mmx:~# date  121611302020.10
Wed Dec 16 11:30:10 UTC 2020
```

3) Write System Time into RTC

Writes the system time set by the previous date command to the RTC device.

```
root@mys-8mmx:~# hwclock -w
```

4) Read Data from RTC and Set to System Time

```
root@mys-8mmx:~# hwclock -r
2020-12-16 11:31:45.723299+00:00
```

5) Keep RTC Time When Power is Off

Press the ON/OFF key, do not disconnect the power, after about ten minutes, press the ON/OFF to power up. Check RTC time and system time.

```
root@mys-8mmx:~# hwclock -r
2020-12-16 11:43:40.683796+00:00
```

6) Make the System Time and the RTC Time Synchronization

```
root@mys-8mmx:~# hwclock -s
root@mys-8mmx:~# date
Wed Dec 16 11:44:23 UTC 2020
```

The RTC time and system time after reboot increased by about 12 minutes compared with the previous settings, indicating that the RTC worked properly. The difference between the test RTC time and the standard time can be extended by 24 hours if the accuracy of the test RTC is required.

2.8. Watchdog

MYS-8MMX chip has 3 watchdog inside, accuracy 0.5 S, maximum time 128 S. There is a watchdog for users，This section demonstrates the use of watchdog and provides some reference procedures for testing watchdog opening, closing, and setting watchdog timeouts.

1) Stop the Watchdog

The watchdog can be closed by writing the "V" character to the watchdog device by command.

```
root@mys-8mmx:~# echo V > /dev/watchdog
```

2) Watchdog Application Testing

```
root@mys-8mmx:~# /unit_tests/Watchdog/wdt_driver_test.out
    ---- Running < /unit_tests/Watchdog/wdt_driver_test.out > test ----
    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 application with a timeout of 4 s, feed the dog once every 1 s:

```
root@mys-8mmx:~# /unit_tests/Watchdog/wdt_driver_test.out  4 1 0

---- Running < /unit_tests/Watchdog/wdt_driver_test.out > test ----

Starting wdt_driver (timeout: 4, sleep: 1, test: ioctl)
Trying to set timeout value=4 seconds
The actual timeout was set to 4 seconds
Now reading back -- The timeout is 4 seconds
```

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

2.9. Power Manager

This section demonstrates the Suspend functionality of Linux power management, allowing development boards to sleep and wake up via external events.

The Linux kernel generally provides three types of suspend: Freeze, Standby, and STR(Suspend to RAM), which can be triggered by writing "freeze", "standby", and "mem" to the "*/sys/power/state*" files in user space. MYS-8MMX only supports suspend to “ram” mode and “mem” mode.

1) View the Mode Supported by the Current Development Board

```
root@mys-8mmx:~# cat /sys/power/state
freeze mem
```

2) Method of Writing in User Space

```
root@mys-8mmx:~# echo "freeze" > /sys/power/state
root@mys-8mmx:~# echo "mem" > /sys/power/state
```

- **Enter mem Power Management Mode**

Take the mem as an example test, after executing the sleep command, the development board sleeps, at this time the debugging serial port can no longer be input, the heartbeat LED stops flashing.

```
root@mys-8mmx:~# echo "mem" > /sys/power/state
[  230.807156] PM: suspend entry (deep)
[  231.594623] Filesystems sync: 0.783 seconds
[  231.598842] [dhd] dhd_pm_callback: action=3, suspend=1, suspend_mode=0
[  231.606210] Freezing user space processes ... (elapsed 0.001 seconds) done.
[  231.614763] OOM killer disabled.
[  231.617988] Freezing remaining freezable tasks ... (elapsed 0.001 seconds)
done.
[  231.626739] printk: Suspending console(s) (use no_console_suspend to deb
ug)
```

- **Wake up through KEY USER**

After entering the sleep,user can wake up normally by pressing the "USER" button:

```
[   72.429671] [dhd] bcmsdh_sdmmc_suspend Enter func->num=2
[   72.429677] [dhd] dhdsdio_suspend Enter
[   72.429680] [dhd] bcmsdh_sdmmc_suspend Exit
[   72.429685] [dhd] bcmsdh_sdmmc_suspend Enter func->num=1
[   72.511998] fec 30be0000.ethernet eth0: Link is Down
[   72.850736] imx_sec_dsim_drv 32e10000.mipi_dsi: wait payload tx done time out
[   73.138742] imx_sec_dsim_drv 32e10000.mipi_dsi: wait pkthdr tx done time out
[   73.138762] panel-toshiba-tc358775 32e10000.mipi_dsi.0: [drm:tc358775_panel_disable] *ERROR* Failed to set display OFF (-16)
[   73.138766] imx_sec_dsim_drv 32e10000.mipi_dsi: panel disable failed: -16
[   73.394717] imx_sec_dsim_drv 32e10000.mipi_dsi: wait pkthdr tx done time out
[   73.394724] panel-toshiba-tc358775 32e10000.mipi_dsi.0: [drm:tc358775_panel_unprepare] *ERROR* Failed to set display OFF (-16)
[   73.682739] imx_sec_dsim_drv 32e10000.mipi_dsi: wait pkthdr tx done time out
[   73.682746] panel-toshiba-tc358775 32e10000.mipi_dsi.0: [drm:tc358775_panel_unprepare] *ERROR* Failed to enter sleep mode (-16)
[   73.747663] PM: suspend devices took 1.320 seconds
[   73.751274] Disabling non-boot CPUs ...
[   73.751870] CPU1: shutdown
[   73.770366] psci: Retrying again to check for CPU kill
[   73.770370] psci: CPU1 killed.
[   73.772472] CPU2: shutdown
[   73.772480] psci: CPU2 killed.
[   73.774446] CPU3: shutdown
[   73.774451] psci: CPU3 killed.
[   73.776121] Enabling non-boot CPUs ...
[   73.776605] Detected VIPT I-cache on CPU1
[   73.776636] GICv3: CPU1: found redistributor 1 region 0:0x00000000388a0000
[   73.776687] CPU1: Booted secondary processor 0x0000000001 [0x410fd034]
[   73.777318] CPU1 is up
[   73.777673] Detected VIPT I-cache on CPU2
[   73.777687] GICv3: CPU2: found redistributor 2 region 0:0x00000000388c0000
[   73.777707] CPU2: Booted secondary processor 0x0000000002 [0x410fd034]
[   73.778074] CPU2 is up
[   73.778437] Detected VIPT I-cache on CPU3
[   73.778451] GICv3: CPU3: found redistributor 3 region 0:0x00000000388e0000
[   73.778471] CPU3: Booted secondary processor 0x0000000003 [0x410fd034]
[   73.778824] CPU3 is up
[   73.855476] [dhd] bcmsdh_sdmmc_resume Enter func->num=2
[   73.855484] [dhd] dhdsdio_resume Enter
[   73.855486] [dhd] bcmsdh_sdmmc_resume Exit
[   73.855745] [dhd] bcmsdh_sdmmc_resume Enter func->num=1
[   73.920806] PM: resume devices took 0.140 seconds
[   74.130902] OOM killer enabled.
[   74.134040] Restarting tasks ... done.
[   74.138528] [dhd] dhd_pm_callback: action=4, suspend=0, suspend_mode=0
[   74.146451] PM: suspend exit
```

At this point, the debugging port can input again, and the heartbeat lamp starts flashing.

3. Peripheral Interface

3.1. GPIO

The testing of GPIO is implemented through the file system sysfs interface. The pin of GPIO5_IO3 is taken as an example to illustrate the use of GPIO.

The GPIO pin of the MYS-8MMX is defined in GPIOX_Y form. Since the GPIOX_Y is converted to the pin number formula :(X-1)*32Y, the GPIO5_IO3 number is :(5-1)*32 3=131.The following describes how to set GPIO under uboot and kernel .

1) Set GPIO under Uboot

- **Enter Uboot Mode**

Press any key to enter uboot mode:

```
Fastboot: Normal
Normal Boot
Hit any key to stop autoboot:  0
u-boot=>
u-boot=>
```

- **Control LED**

Set LED lights on and off by uboot command，turn off the led：

```
u-boot=> gpio set 131
gpio: pin 131 (gpio 131) value is 1
```

Turn on the led：

```
u-boot=> gpio clear 131
gpio: pin 131 (gpio 131) value is 0
```

2) Set GPIO under Sysfs

When operating the GPIO under the file system, you must first check whether the GPIO is occupied by other drivers, and if temporarily used, the subsequent operation will report a busy warning. Take the GPIO of the extended interface (J16) as an example and use a multimeter to measure the voltage. For example :(SAI2_TXD,GPIO4_IO26,122).

- **Export GPIO**

```
root@mys-8mmx:~# echo 122 > /sys/class/gpio/export
```

The gpio122 directory is generated in the /sys/class/gpio/ directory after gpio122 export is successful.

- **Set/View the GPIO Value**

Set as Output:

```
root@mys-8mmx:~#  echo "out" > /sys/class/gpio/gpio122/direction
```

Set as Input:

```
root@mys-8mmx:~# echo "in" > /sys/class/gpio/gpio122/direction
```

View the Direction of GPIO:

```
root@mys-8mmx:/sys/class/gpio# cat /sys/class/gpio/gpio122/direction
out
```

Return “in” for input and “out” for output.

- **Set/View the GPIO Value**

Set Output Low:

```
root@mys-8mmx:~# echo "0" > /sys/class/gpio/gpio122/value
```

Set Output High:

```
root@mys-8mmx:~# echo "1" > /sys/class/gpio/gpio122/value
```

View the GPIO Value:

```
root@mys-8mmx:~# cat /sys/class/gpio/gpio122/value
1
```

You can see that the output level of GPIO4_IO26 is high. You can use a multimeter to measure the GPIO4_IO26 pin of j16. You can see that the voltage is about 3.3V.

3.2. LED

The Linux system provides a separate subsystem to facilitate the operation of LED devices from user space. The subsystem provides an interface for LED devices in the form of files. These interfaces are located in the */sys/class/leds* directory. In the list of hardware resources, we have listed all the LEDs on the device. Let's test a LED by reading and writing sysfs from the command. The following commands are generic commands and are general methods of controlling LEDs .

1) View the LEDs

The directory where leds are operated is */sys/class/leds*. The contents of the directory are as follows:

```
root@mys-8mmx:~# ls  /sys/class/leds/
cpu  mmc0::  mmc1::  mmc2::  user1  user2
```

2) Test a LED

- **Get the Status of the LED**

Read user LED status,where 0 means the LED is on and 1 means the LED is off.

```
root@mys-8mmx:~# cat  /sys/class/leds/user2/brightness
255
```

The above results show that the current user light LED is on.

- **Turn off the LED**

```
root@mys-8mmx:~# echo 0 > /sys/class/leds/user2/brightness
```

- **Turn on the LED**

```
root@mys-8mmx:~# echo 1 > /sys/class/leds/user2/brightness
```

- **Trigger the LED**

After starting the " heartbeat" trigger mode, the LED flashes at a default heartbeat.

```
root@mys-8mmx:~# echo heartbeat > /sys/class/leds/user2/trigger
```

3.3. Key

The "*/dev/input/eventxx*" devices in Linux system can be used to easily debug input devices such as mice, keyboards, touchpads, etc.This section focuses on testing the key. We can use the "evtest" command to see if the keys work normally.MYS-8MMX has three buttons, and K1 is the on/off button,K2 is the system reset button, K3 is the user key that has been configured in the device tree.

1) Device Tree Node

Open the supporting device tree file *myb-imx8mm-base.dts*, and you can see the corresponding GPIO-Keys node:

```
gpio-keys {
                compatible = "gpio-keys";
                pinctrl-names = "default";
                pinctrl-0 = <&pinctrl_gpio_key>;

                user {
                        label = "User";
                        gpios = <&gpio3 19 GPIO_ACTIVE_LOW>;
                        gpio-key,wakeup;
                        linux,code = <KEY_1>;
                };
          };
```

2) Test Keys

- **View Information of the Input Device Event**

```
root@mys-8mmx:~# evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      30370000.snvs:snvs-powerkey
/dev/input/event1:      gpio-keys
```

```
/dev/input/event2:      bd718xx-pwrkey
Select the device event number [0-2]:
```

The above results show that the corresponding device event for GPIO-Keys is “event1” .

- **Test key**

Select event 1 and press the button K3, the serial terminal will print out the following information:

```
Select the device event number [0-2]: 1
Input driver version is 1.0.1
Input device ID: bus 0x19 vendor 0x1 product 0x1 version 0x100
Input device name: "gpio-keys"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 2 (KEY_1)
Properties:
Testing ... (interrupt to exit)
Event: time 1608127834.064981, type 1 (EV_KEY), code 2 (KEY_1), value 1
Event: time 1608127834.064981, -------------- SYN_REPORT ------------
Event: time 1608127834.216987, type 1 (EV_KEY), code 2 (KEY_1), value 0
Event: time 1608127834.216987, -------------- SYN_REPORT ------------
Event: time 1608127835.300985, type 1 (EV_KEY), code 2 (KEY_1), value 1
Event: time 1608127835.300985, -------------- SYN_REPORT ------------
Event: time 1608127835.416981, type 1 (EV_KEY), code 2 (KEY_1), value 0
Event: time 1608127835.416981, -------------- SYN_REPORT ------------
```

Each time you press the K3 button, the current terminal prints out the current event code value, that is, the button is normal.

3.4. USB

The MYS-8MMX device has two USB2.0 interfaces, one is a microusb interface, the other USB2.0 port is converted into two channels through the extended chip.

This section through the relevant command or hot plug, USB HUB verify the feasibility of USB Host drive, to achieve read-write U disk function, usb enumeration function.

1) Check the Kernel Message of USB

- **View the USB Device Information**

Connect the U disk to the USB Host interface of the development board, and the kernel log is as follows:

```
root@mys-8mmx:~# [  775.418981] usb 1-1.1: new high-speed USB device number 3 using ci_hdrc
[  775.548934] usb-storage 1-1.1:1.0: USB Mass Storage device detected
[  775.556003] scsi host0: usb-storage 1-1.1:1.0
[  776.565489] scsi 0:0:0:0: Direct-Access     Kingston DataTraveler 3.0     PQ: 0 ANSI: 6
[  776.576171] sd 0:0:0:0: [sda] 60437492 512-byte logical blocks: (30.9 GB/28.8 GiB)
[  776.584298] sd 0:0:0:0: [sda] Write Protect is off
[  776.589526] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[  776.625823]  sda: sda1
[  776.631144] sd 0:0:0:0: [sda] Attached SCSI removable disk
```

From the above information it can be concluded that the device to be assigned is sda1.

2) Mount and Read/Write the USB Flash Disk

Since the file system has added added to the automatic mount function, as long as the U disk has been formatted format, it can be automatically mounted.

- **View Mount Points**

```
root@mys-8mmx:~# cat /proc/mounts | grep sda*
cgroup2 /sys/fs/cgroup/unified cgroup2 rw,nosuid,nodev,noexec,relatime,nsdele
gate 0 0
/dev/sda1 /run/media/sda1 vfat rw,relatime,gid=6,fmask=0007,dmask=0007,allo
w_utime=0020,codepage=437,iocharset=iso8859-1,shortname=mixed,errors=rem
ount-ro 0 0
```

- **Read USB Flash Disk**

```
root@mys-8mmx:~# echo "myir udisk test" > /run/media/sda1/test_file.txt
```

- **Write USB Flash Disk**

```
root@mys-8mmx:~# cat /run/media/sda1/test_file.txt
myir udisk test
```

After the file is written, the sync command needs to be executed to ensure that the data is fully written to the USB disk before the device can be unmounted.

3) Files preservation after power off

Disk file writing involves the concept of cache, which is divided into full cache, line cache and no cache. The default io is full cache, which is written to disk file only when the cache data is full. So in the process of writing a file, if the power is suddenly cut off, this will make the content written to the file lost. When reboot is used, however, the system helps write buffer data to data, and the command line can force sync data to disk.

3.5. Display

The MYS-8MMX device has a 4-channel MIPI-DSI, and converts the DSI signal into a HDMI signal through a LT8912.Then LVDS signal is generated by scaling HDMI signal and default is a 7-inch screen 1024 x600 resolution display.The image display terminal supports wayland and xwayland (x 11 wayland) and does not support FB mode.

1) Gets the current connection information

- **Gets the current connector type**

```
root@mys-8mmx:~# for p in /sys/class/drm/*/status; do con=${p%/status}; ec
ho -n "${con#*/card?-}: "; cat $p; done
DSI-1: connected
```

From above you can know that the current DSI connection.

- **Gets the current connection state**

```
root@mys-8mmx:~# cat /sys/kernel/debug/dri/0/state
plane[31]: plane-0
        crtc=crtc-0
        fb=38
                allocated by = weston
                refcount=2
                format=XR24 little-endian (0x34325258)
                modifier=0x0
                size=1024x599
                layers:
                        size[0]=1024x599
                        pitch[0]=4096
                        offset[0]=0
                        obj[0]:
                                name=0
                                refcount=4
                                start=00100257
                                size=2453504
                                imported=no
        crtc-pos=1024x599+0+0
        src-pos=1024.000000x599.000000+0.000000+0.000000
        rotation=1
        normalized-zpos=0
        color-encoding=ITU-R BT.601 YCbCr
        color-range=YCbCr limited range
crtc[33]: crtc-0
        enable=1
        active=1
        self_refresh_active=0
        planes_changed=1
        mode_changed=0
        active_changed=0
        connectors_changed=0
        color_mgmt_changed=0
        plane_mask=1
        connector_mask=1
        encoder_mask=1
        mode: "1024x599": 60 50000 1024 1048 1164 1324 599 601 604 628
0x48 0xa
connector[35]: DSI-1
        crtc=crtc-0
        self_refresh_aware=0
```

Gets the status of the current connector：

```
root@mys-8mmx:~# cat /sys/class/drm/card0-DSI-1/status
connected
```

The current connector is normal.

- **Gets the current display mode**

```
root@mys-8mmx:~# cat /sys/class/drm/card0-DSI-1/modes
1024x599
```

The current resolution is 1024*599.

2) LVDS Display Test

The MYS-8MMX device adapts to MYIR's LVDS screen model of MY-LVDS070CV11.the screen is connected to the J6 LVDS interface of the backplane through a 50 pin flexible cable，Then see if the output is normal.

You can then manually input commands to generate a test screen to see if the output is normal:

```
root@mys-8mmx:~# gst-launch-1.0 videotestsrc pattern=18 ! waylandsink
Setting pipeline to PAUSED ...
Pipeline is PREROLLING ...
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
```

You can see the screen can normally display the ball animation, which indicates that the display is normal.

3.6. Backlight

This routine is mainly to test the brightness of the backlight. It can query and adjust the brightness of the backlight by operating the corresponding file of the backlight driver in “*/sys/class/backlight*” directory.

1) View current backlight level

- **The SYSFS Directory of Backlight Driver**

```
root@mys-8mmx:~# cd /sys/class/backlight/lvds_backlight@0
root@mys-8mmx:/sys/class/backlight/lvds_backlight@0# ls
actual_brightness  bl_power  brightness  consumers  device  max_brightness
power  scale  subsystem  suppliers  type  uevent
```

- **View current backlight level**

```
root@mys-8mmx:/sys/class/backlight/lvds_backlight@0# cat brightness
80
```

- **View maximum backlight level**

```
root@mys-8mmx:/sys/class/backlight/lvds_backlight@0# cat max_brightness
100
```

From the above, the maximum backlight level is 100, the range is 0-100.

2) Adjust backlight level

if you need to adjust the backlight level, write the corresponding need to adjust the backlight level, but not beyond the maximum level.

- **Turn off Backlight**

If you write'0' to the brightness parameter, you turn off the backlight:

```
root@mys-8mmx:/sys/class/backlight/lvds_backlight@0# echo 0 > brightness
```

- **Adjust Backlight**

Write a positive integer of 0-100 directly to the brightness parameter:

```
root@mys-8mmx:/sys/class/backlight/lvds_backlight@0# echo 60 > brightness
```

3.7. Touch Panel

The general touch functions include capacitive touch and resistive touch. The hardware of MYS-8MMX series development board does not support resistance touch at present, but supports capacitive touch.Please refer to the LVDS screen in Table 1-1 for the test accessories. Users can purchase accessories by themselves according to actual requirements.The capacitive screen is sensitive in use and seldom has problems. In addition, the capacitive touch screen does not need to be accurate. The following is a simple example of testing the touch function of capacitive screen through the evtest command.

1) Touch screen connection

Connect MY-LVDS070CV11 LVDS to the development board according to section 3.5.

2) Touch test with evtest command

The terminal executes "evtest" to enter the test interface. Select the test peripheral as the touch screen, where the default is input interrupt 1. Select "1" in the test interface and press enter to start the test:

```
root@mys-8mmx:~#  evtest
No device specified, trying to scan all of /dev/input/event*
Available devices:
/dev/input/event0:      30370000.snvs:snvs-powerkey
/dev/input/event1:      generic ft5x06 (00)
/dev/input/event2:      gpio-keys
/dev/input/event3:      bd718xx-pwrkey
Select the device event number [0-3]: 1
Input driver version is 1.0.1
Input device ID: bus 0x18 vendor 0x0 product 0x0 version 0x0
Input device name: "generic ft5x06 (00)"
Supported events:
  Event type 0 (EV_SYN)
  Event type 1 (EV_KEY)
    Event code 330 (BTN_TOUCH)
  Event type 3 (EV_ABS)
    Event code 0 (ABS_X)
      Value    337
      Min        0
      Max     1023
    Event code 1 (ABS_Y)
      Value    430
      Min        0
      Max      599
    Event code 47 (ABS_MT_SLOT)
      Value      0
      Min        0
      Max        4
    Event code 53 (ABS_MT_POSITION_X)
      Value      0
      Min        0
      Max     1023
    Event code 54 (ABS_MT_POSITION_Y)
      Value      0
      Min        0
      Max      599
    Event code 57 (ABS_MT_TRACKING_ID)
      Value      0
      Min        0
      Max    65535
Properties:
  Property type 1 (INPUT_PROP_DIRECT)
Testing ... (interrupt to exit)
Event: time 1608131697.757201, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value 97
```

```
Event: time 1608131697.757201, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 519
Event: time 1608131697.757201, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 323
Event: time 1608131697.757201, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 1
Event: time 1608131697.757201, type 3 (EV_ABS), code 0 (ABS_X), value 519
Event: time 1608131697.757201, type 3 (EV_ABS), code 1 (ABS_Y), value 323
Event: time 1608131697.757201, -------------- SYN_REPORT ------------
Event: time 1608131697.805391, type 3 (EV_ABS), code 53 (ABS_MT_POSITION_X), value 521
Event: time 1608131697.805391, type 3 (EV_ABS), code 54 (ABS_MT_POSITION_Y), value 322
Event: time 1608131697.805391, type 3 (EV_ABS), code 0 (ABS_X), value 521
Event: time 1608131697.805391, type 3 (EV_ABS), code 1 (ABS_Y), value 322
Event: time 1608131697.805391, -------------- SYN_REPORT ------------
Event: time 1608131697.864142, type 3 (EV_ABS), code 57 (ABS_MT_TRACKING_ID), value -1
Event: time 1608131697.864142, type 1 (EV_KEY), code 330 (BTN_TOUCH), value 0
Event: time 1608131697.864142, -------------- SYN_REPORT ------------

………
```

From the above, when you click on the LVDS screen, the screen will print out the key coordinates, which indicates that the touch is normal.

3.8. MIPI-CSI

The MYS-8MMX device has a 4-channel MIPI-CSI,this section tests for MY-CAM003M cameras, up to 500 W pixels,please refer to the CSI cameras in Table 1-1 for the test accessories.

1) Check the driver by i2cdetect command

You can find the corresponding address of the device through the corresponding I2C schematic or device tree node. Camera's address is 0x3C：

```
root@mys-8mmx:~# i2cdetect -y 1
     0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:          -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- UU -- -- -- -- -- -- -- -- -- UU -- -- --
40: -- -- -- -- -- -- -- -- UU UU UU 4b -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- 6a -- -- -- -- --
70: -- -- -- -- -- -- -- --
```

The UU, in the 3C address is detected, which represents the normal load of the driver.

2) View the Device Node

```
root@mys-8mmx:~#  ls -l /dev/video*
crw-rw---- 1 root video 81, 0 Dec 16 15:18 /dev/video0
```

3) View the Format Supported by the Device

```
root@mys-8mmx:~# v4l2-ctl -D -d /dev/video0 --list-formats-ext
```

```
Driver Info:
        Driver name      : mx6s-csi
        Card type        : i.MX6S_CSI
        Bus info         : platform:32e20000.csi1_bridge
        Driver version   : 5.4.3
        Capabilities     : 0x84200001
                Video Capture
                Streaming
                Extended Pix Format
                Device Capabilities
        Device Caps      : 0x04200001
                Video Capture
                Streaming
                Extended Pix Format
ioctl: VIDIOC_ENUM_FMT
        Type: Video Capture

        [0]: 'YUYV' (YUYV 4:2:2)
                Size: Discrete 640x480
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 320x240
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x480
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1280x720
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1920x1080
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 2592x1944
                        Interval: Discrete 0.067s (15.000 fps)
```

- Driver name：Driver name
- Ioctl：VIDIOC_ENUM_FMT：Support format

4) How to Make a Camera Preview

Execute the following command to preview:

```
root@mys-8mmx:~# gst-launch-1.0 v4l2src device=/dev/video0 ! 'video/x-raw,
width=720,height=480,framerate=30/1' ! glimagesink
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Got context from element 'sink': gst.gl.GLDisplay=context, gst.gl.GLDisplay=(Gs
tGLDisplay)"\(GstGLDisplayWayland\)\ gldisplaywayland0";
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[   24.342573] ov5640_mipi 1-003c: s_stream: 1
```

- Device: Specifies the device node
- Video/x-raw：Video format displayed
- Width：Display width, need to match system support width
- Height：High display
- Framerate： Frame rate, need to match frame rate with system support
- Glimagesink：Display terminal

5) Save camera data locally

- **gst-launch Command Syntax**

You can use the gst-launch tool to preview and save the code, and execute the following commands on the terminal interface to see the command syntax:

```
root@mys-8mmx:~# gst-launch-1.0 -help
```

```
Usage:
  gst-launch-1.0 [OPTION?] PIPELINE-DESCRIPTION
Help Options:
  -h, --help                          Show help options
  --help-all                          Show all help options
  --help-gst                          Show GStreamer Options
Application Options:
  -t, --tags                          Output tags (also known as metadata)
  -c, --toc                           Output TOC (chapters and editions)
  -v, --verbose                       Output status information and property no
tifications
  -q, --quiet                         Do not print any progress information
  -m, --messages                      Output messages
  -X, --exclude=PROPERTY-NAME         Do not output status information for
the specified property if verbose output is enabled (can be used multiple tim
es)
  -f, --no-fault                      Do not install a fault handler
  -e, --eos-on-shutdown               Force EOS on sources before shutting th
e pipeline down
  --version                           Print version information and exit
```

- **Coding and Saving of H.264 Format**

Execute the following command to encode and save the video in H.264 format:

```
root@mys-8mmx:~# gst-launch-1.0  -e  v4l2src device=/dev/video0 io-mode=
2 ! video/x-raw,format=YUY2,width=320,height=240,frmaerate=30/1 ! vpuenc_h
264 ! queue ! h264parse ! qtmux ! filesink location=uvc_h264.mp4
Setting pipeline to PAUSED ...
====== VPUENC: 4.5.4 build on Nov 24 2020 02:09:55. ======
        wrapper: 3.0.0 (VPUWRAPPER_ARM64_LINUX Build on Nov 23 2020 0
9:58:25)
        vpulib: 1.1.1
        firmware: 1.1.1.43690
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[  296.502400] ov5640_mipi 1-003c: s_stream: 1
^Chandling interrupt.
Interrupt: Stopping pipeline ...
EOS on shutdown enabled -- Forcing EOS on th[  314.879756] ov5640_mipi 1
-003c: s_stream: 0
e pipeline
Waiting for EOS...
Got EOS from element "pipeline0".
EOS received - stopping pipeline...
Execution ended after 0:00:19.045536750
Setting pipeline to PAUSED ...
Setting pipeline to READY ...
Setting pipeline to NULL ...
Freeing pipeline ...
root@mys-8mmx:~#
```

Partial parameter description:

- v4l2src：Video capture plug-in
- device：Configure the collected device node, readable and writable flag
- video/x-raw,format:Configure the video format, length, width and frame rate of the acquisition

Play H.264 video saved locally：

```
root@mys-8mmx:~# gst-launch-1.0 filesrc location=uvc_h264.mp4 typefind=tru
e ! video/quicktime ! qtdemux ! queue ! vpudec ! glimagesink
Setting pipeline to PAUSED ...
====== VPUDEC: 4.5.4 build on Nov 24 2020 02:09:55. ======
        wrapper: 3.0.0 (VPUWRAPPER_ARM64_LINUX Build on Nov 23 2020 09:58:27)
        vpulib: 1.1.1
        firmware: 1.1.1.0
Pipeline is PREROLLING ...
Got context from element 'sink': gst.gl.GLDisplay=context, gst.gl.GLDisplay=(GstGLDisplay)"\(GstGLDisplayWayland\)\ gldisplaywayland0";
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
Got EOS from element "pipeline0".
Execution ended after 0:00:16.268494750
Setting pipeline to PAUSED ...
Setting pipeline to READY ...
Setting pipeline to NULL ...
Total showed frames (446), playing for (0:00:16.269153875), fps (27.414).
Freeing pipeline ...
```

- **Coding and Saving of VP8 Format**

Execute the following command to encode and save the video in VP8 format:

```
root@mys-8mmx:~# gst-launch-1.0  -e  v4l2src device=/dev/video0 io-mode=2 ! video/x-raw,format=YUY2,width=640,height=480,frmaerate=30/1 ! vpuenc_vp8 ! queue ! qtmux ! filesink location=uvc_vp8.kmv
Setting pipeline to PAUSED ...
====== VPUENC: 4.5.4 build on Nov 24 2020 02:09:55. ======
        wrapper: 3.0.0 (VPUWRAPPER_ARM64_LINUX Build on Nov 23 2020 09:58:25)
        vpulib: 1.1.1
        firmware: 1.1.1.43691
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[  845.175062] ov5640_mipi 1-003c: s_stream: 1
^Chandling interrupt.
```

```
Interrupt: Stopping pipeline ...
EOS on shutdown enabled -- Forcing EOS on th[  867.366251] ov5640_mipi 1
-003c: s_stream: 0
e pipeline
Waiting for EOS...
Got EOS from element "pipeline0".
EOS received - stopping pipeline...
Execution ended after 0:00:22.935927500
Setting pipeline to PAUSED ...
Setting pipeline to READY ...
Setting pipeline to NULL ...
Freeing pipeline ...
```

Play VP8 video saved locally：

```
root@mys-8mmx:~# gst-launch-1.0 filesrc location=uvc_vp8.kmv typefind=true
 ! video/quicktime ! qtdemux ! queue ! vpudec ! glimagesink
Setting pipeline to PAUSED ...
====== VPUDEC: 4.5.4 build on Nov 24 2020 02:09:55. ======
        wrapper: 3.0.0 (VPUWRAPPER_ARM64_LINUX Build on Nov 23 2020 0
9:58:27)
        vpulib: 1.1.1
        firmware: 1.1.1.0
Pipeline is PREROLLING ...
Got context from element 'sink': gst.gl.GLDisplay=context, gst.gl.GLDisplay=(Gs
tGLDisplay)"\(GstGLDisplayWayland\)\ gldisplaywayland0";
Pipeline is PREROLLED ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
Got EOS from element "pipeline0".
Execution ended after 0:00:17.694472250
Setting pipeline to PAUSED ...
Setting pipeline to READY ...
Setting pipeline to NULL ...
Total showed frames (485), playing for (0:00:17.695104875), fps (27.409).
```

```
Freeing pipeline ...
```

- **Preview and Save**

Encoded and saved in H.264 format:

```
root@mys-8mmx:~# gst-launch-1.0  -e v4l2src device=/dev/video0 io-mode=2
 ! video/x-raw,format=YUY2,width=320,height=240,frmaerate=30/1 ! tee name
=t !  queue ! glimagesink t. ! queue ! vpuenc_h264 ! queue ! h264parse ! qt
mux ! filesink location=uvc_h264.mp4
Setting pipeline to PAUSED ...
====== VPUENC: 4.5.4 build on Nov 24 2020 02:09:55. ======
        wrapper: 3.0.0 (VPUWRAPPER_ARM64_LINUX Build on Nov 23 2020 0
9:58:25)
        vpulib: 1.1.1
        firmware: 1.1.1.43690
Pipeline is live and does not need PREROLL ...
Got context from element 'sink': gst.gl.GLDisplay=context, gst.gl.GLDisplay=(Gs
tGLDisplay)"\(GstGLDisplayWayland\)\ gldisplaywayland0";
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[  219.574486] ov5640_mipi 1-003c: s_stream: 1
```

Encoded and saved in VP8 format:

```
root@mys-8mmx:~# gst-launch-1.0  -e  v4l2src device=/dev/video0 io-mode=
2 ! video/x-raw,format=YUY2,width=640,height=480,frmaerate=30/1 ! tee name
=t !  queue ! glimagesink t. ! queue ! vpuenc_vp8 ! queue ! qtmux ! filesink
location=uvc_vp8.kmv
Setting pipeline to PAUSED ...
====== VPUENC: 4.5.4 build on Nov 24 2020 02:09:55. ======
        wrapper: 3.0.0 (VPUWRAPPER_ARM64_LINUX Build on Nov 23 2020 0
9:58:25)
        vpulib: 1.1.1
        firmware: 1.1.1.43690
Pipeline is live and does not need PREROLL ...
```

```
Got context from element 'sink': gst.gl.GLDisplay=context, gst.gl.GLDisplay=(GstGLDisplay)"\(GstGLDisplayWayland\)\ gldisplaywayland0";
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[  378.743888] ov5640_mipi 1-003c: s_stream: 1
```

- **Web Camera**

Preview and encode in H264 format and then transfer using UDP:

```
gst-launch-1.0 v4l2src device=/dev/video0 io-mode=2 ! video/x-raw,format=YUY2,width=640,height=480,framerate=30/1 ! tee name=t ! queue ! waylandsink t. ! queue ! vpuenc_h264 ! rtph264pay ! udpsink host=192.168.30.178 port=1234
```

Then play at the other end：

```
gst-launch-1.0 udpsrc port=1234 !  application/x-rtp ! rtph264depay ! vpudec ! glimagesink
```

3.9. M.2

MYS-8MMX have a M.2 interface, you can connect to the SSD hard disk of the M.2 interface, and then you can read and write the SSD device.

Note that the newly purchased SSD card does not format the disk, so it will not mount automatically. It needs to be formatted with fsck.fat/fsck.ext2/fsck.ext3,fsck.ext4 commands.

1) How to view SSD

- **Check Disk using df Command**

```
root@mys-8mmx:~# df -h
Filesystem       Size  Used Avail Use% Mounted on
/dev/root        3.5G  2.2G  1.1G  68% /
devtmpfs         649M  4.0K  649M   1% /dev
tmpfs            972M     0  972M   0% /dev/shm
tmpfs            972M  9.0M  963M   1% /run
tmpfs            972M     0  972M   0% /sys/fs/cgroup
tmpfs            972M  4.0K  972M   1% /tmp
tmpfs            972M  204K  972M   1% /var/volatile
tmpfs            195M  4.0K  195M   1% /run/user/0
/dev/nvme0n1     110G  561M  104G   1% /run/media/nvme0n1
/dev/mmcblk2p1    84M   35M   49M  42% /run/media/mmcblk2p1
```

- ➢ /dev/nvme0n1: SSD hard drive ,110 G size

2) How to format SSD partitions

Execute the fsck.ext4 command to format the partition into ext4 format. If it has been mounted automatically, unmount it before formatting.

```
root@mys-8mmx:~# umount /run/media/nvme0n1/
root@mys-8mmx:~# mkfs.ext4 /dev/nvme0n1
```

```
mke2fs 1.45.3 (14-Jul-2019)
/dev/nvme0n1 contains a ext4 file system
        last mounted on /run/media/nvme0n1 on Wed Dec 16 17:20:25 2020
Proceed anyway? (y,N) y
Discarding device blocks: done
Creating filesystem with 29305206 4k blocks and 7331840 inodes
Filesystem UUID: 844346ae-bf61-4257-af0d-219557d2cb08
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (131072 blocks): done
Writing superblocks and filesystem accounting information: done
```

3) How to read and write SSD

If it has been formatted successfully, the device is automatically mounted when the system is rebooted. Below we will demonstrate how to view, write, and read files.

- **View Mount Points**

```
root@mys-8mmx:~# cat /proc/mounts | grep nvme
/dev/nvme0n1 /run/media/nvme0n1 ext4 rw,relatime 0 0
```

- **Writing File**

```
root@mys-8mmx:~#  echo "myir nvme test" >> /run/media/nvme0n1/nvme_test.txt
```

- **Reading FIle**

```
root@mys-8mmx:~# cat /run/media/nvme0n1/nvme_test.txt

myir nvme test
```

4. Network Interface

The MYS-8MMX development board includes a gigabit Ethernet interface and a WIFI/Bluetooth Combo Module(AP6256). The configuration of these two network devices is described below.

4.1. Ethernet

Under Linux, there are many tools for network configuration, such as net-tools, Iproute2, Systemd-Networkd, Network Manager and Connman, etc. All of these can be selected according to actual needs during system customization. Here, several commonly used methods of Ethernet manual temporary configuration and automatic permanent configuration are introduced.

1) Configure Ethernet IP addresses Manually and Temporarily

- **Use ifconfig to Configure the Network Manually**

check the network device information through ifconfig command as follows:

```
root@mys-8mmx:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 86:9d:1a:77:77:4c
          inet addr:192.168.40.202  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::849d:1aff:fe77:774c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:720 errors:0 dropped:0 overruns:0 frame:0
          TX packets:82 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:63840 (62.3 KiB)  TX bytes:10294 (10.0 KiB)
```

Eth0 is the actual Ethernet device,where HWaddr represents the MAC address of the system, the code is read in the following order, if the first is not set, then look for the next:

- uboot incoming parameters(fec.macddr)
- MAC written in the device tree file
- MAC written to flash or fuse

- MAC written to FEC mac registers under uboot
- Random MAC address

Eth0 is given the following command to manually configure the IP address 192.168.40.200：

```
root@mys-8mmx:~# ifconfig eth0 192.168.40.200 netmask 255.255.255.0 up
root@mys-8mmx:~# route add default gw 192.168.40.1
```

The command above manually configured eth0 with the IP address of 192.168.40.200, the subnet mask of 255.255.255.0, and the default configured broadcast address of 192.168.40.255, and activated with the up parameter, as shown below:

```
root@mys-8mmx:~# ifconfig eth0
eth0       Link encap:Ethernet  HWaddr 86:9d:1a:77:77:4c
           inet addr:192.168.40.200  Bcast:192.168.40.255  Mask:255.255.255.0
           inet6 addr: fe80::849d:1aff:fe77:774c/64 Scope:Link
           UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
           RX packets:1248 errors:0 dropped:0 overruns:0 frame:0
           TX packets:119 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:107725 (105.2 KiB)  TX bytes:15027 (14.6 KiB)
```

- **Configure the Network Manually by iprout2**

The ifconfig command that sets the IP address manually can also be substituted using “ip addr” and “ip link”, for more information about iprout2, please refre to https://wiki.linuxfoundation.org/networking/iproute2 for detail.

```
root@mys-8mmx:~#  ip addr flush dev eth0
root@mys-8mmx:~#  ip addr add 192.168.40.101/24 brd + dev eth0
root@mys-8mmx:~#  ip link set eth0 up
root@mys-8mmx:~# route add default gw 192.168.40.1
```

If an IP address has been previously configured, the IP address configured using “ip addr add” will become the secondary address, so use “ip addr flush” to clear the previous address before configuring and activating it. Once configured, view the eth0 information from the IP Addr Show command as follows:

```
root@mys-8mmx:~# ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state U
P group default qlen 1000
    link/ether 86:9d:1a:77:77:4c brd ff:ff:ff:ff:ff:ff
    inet 192.168.40.101/24 brd 192.168.40.255 scope global eth0
       valid_lft forever preferred_lft forever
```

2) Configure Ethernet Automatically and Permanently

IP addresses configured through the “ifconfig” and “ip” commands are lost when power is lost, and if you need to make IP addresses permanent, you need to modify the corresponding configuration files of the network management tool.

MYS-8MMX use systemd-networkd services to manage the network, default automatic access ip, the following describes how to configure static ip addresses.

- **View /lib/systemd/network**

```
root@mys-8mmx:/lib/systemd/network# ls -l
total 20
-rw-r--r-- 1 root root 645 Nov 18 07:47 80-container-host0.network
-rw-r--r-- 1 root root 718 Nov 18 07:47 80-container-ve.network
-rw-r--r-- 1 root root 704 Nov 18 07:47 80-container-vz.network
-rw-r--r-- 1 root root 113 Nov 18 08:07 80-wired.network
-rw-r--r-- 1 root root 441 Nov 18 07:47 99-default.link
```

- **View 80-wired.network**

```
root@mys-8mmx:/lib/systemd/network# cat 80-wired.network
[Match]
Name=en* eth*
KernelCommandLine=!nfsroot

[Network]
DHCP=yes

[DHCP]
RouteMetric=10
```

```
ClientIdentifier=mac
```

The configuration file above will configure the network matching en* and eth*. If the kernel boot parameter does not contain the nfsroot parameter, then the DHCP will automatically configure the IP address, gateway, DNS and other information for the matched network card. See the following connection for detailed configuration parameters：
https://www.freedesktop.org/software/systemd/man/systemd.network.html.

The main network file must have the extension xx-xxxx.network; other extensions are ignored.The xx-xxxx.network files are read from the files located in the system network directories */lib/systemd/network*， the volatile runtime network directory */run/systemd/network/* and the local administration network directory */etc/systemd/network*.The kernel read configuration priority is / etc/>/ run/etc/>/ lib/. Where the network card file format is a digital+name. network. Here's how to set static ip.

- **Create/ etc/systemd/network/10-static-eth0.network File**

```
[Match]
Name=eth0

[Network]
Address=192.168.40.202/24
Gateway=192.168.40.1
DNS=8.8.8.8
```

- **Restart the Network**

```
root@mys-8mmx:~#  systemctl  restart  systemd-networkd
```

4.2. Wi-Fi

This section mainly introduces the configuration and use of Wi-Fi under Linux. Generally, the Wi-Fi module can support two working modes, STA mode and AP mode, and some devices also support STA and AP mode to work simultaneously. The STA mode allows the device to connect to an external Wi-Fi hotspot, and the AP mode turns the device into a Wi-Fi hotspot for other devices to connect to.

MYS-8MMX equipped with WIFI/Bluetooth Combo Module(AP6256),currently, STA and AP are not supported to work simultaneously. The driver corresponding to the AP6256 Wi-Fi module is:

```
root@mys-8mmx:~# lsmod
Module                  Size  Used by
bcmdhd               1286144  0
............
```

During the driver load, the Wi-Fi firmware located at */lib/firmware/brcm* will be loaded into the module. After the Wi-Fi module driver is loaded successfully, the network node “wlan0” of the Wi-Fi device is generated, as shown below:

```
root@mys-8mmx:~# ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr 18:93:7f:74:5e:32
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

1) Manually connect to WiFi hotspot for STA mode

Now try manually connecting to a nearby Wi-Fi hotspot "MYIR_TECH", a WPA2 encrypted Wi-Fi hotspot with the password myir@2016.

- **Ensure wlan0 network device is active** rfkill is used to manage the software wifi/bt, so the first step is to query the current state, if it is block (closed), and then execute the unblock wifi command to open the wifi.

```
root@mys-8mmx:~# rfkill list
0: phy0: Wireless LAN
        Soft blocked: no
        Hard blocked: no
1: brcmfmac-wifi: Wireless LAN
        Soft blocked: no
        Hard blocked: no
root@mys-8mmx:~#  ifconfig wlan0 up
[ 1030.730271] [dhd] dhd_open: Enter wlan0
[ 1030.734241] [dhd] dhd_open : no mutex held. set lock
[ 1030.739396] [dhd]
[ 1030.739396] Dongle Host Driver, version 100.10.545.12 (r826445-20200708-1)
[ 1030.748342] [dhd-wlan0] wl_android_wifi_on : in g_wifi_on=0
[ 1030.753998] [dhd] wifi_platform_set_power = 1, delay: 200 msec
...........................
```

- **Scan the Wi-Fi hotspots nearby**

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

```
root@mys-8mmx:~# iw dev wlan0 scan | grep SSID
[ 1072.063423] [dhd-wlan0] wl_run_escan : LEGACY_SCAN sync ID: 0, bssidx: 0
        SSID: MI8
        SSID: MYIR_TECH
        SSID: HP-Print-3C-LaserJet Pro MFP
        SSID: HUAWEI_B316_CA1C_Guest
        SSID: HUAWEI_B316_CA1C
        SSID:
```

- **Enable wpa_supplicant server**

The wpa_supplicant tool contains two programs: wpa_supplicant and wpa_cli, in which the wpa_supplicant program runs as a server in the background and serves the request of the wpa_cli client, thus realizing the configuration connection of the WiFi. The wpa_supplicant required before using the wpa_cli client has been run as a server.as shown below:

```
root@mys-8mmx:~# wpa_supplicant -B -iwlan0 -c /etc/wpa_supplicant.conf
Successfully initialized wpa_supplicant
root@mys-8mmx:~# [ 2413.498428] [dhd-wlan0] wl_run_escan : LEGACY_SCAN
 sync ID: 1, bssidx: 0
root@mys-8mmx:~#
root@mys-8mmx:~# [ 2419.633812] [dhd-wlan0] wl_run_escan : LEGACY_SCAN
 sync ID: 2, bssidx: 0
```

- -B : Run daemons in the background
- -D : Driver name
- -c : The path to the configuration file
- -i : Wi-Fi network device node

- **Connection Hotspot**

The wpa_cli and wpa_supplicant C/S architecture is used to complete the hot spot connection, if the hot spot does not set the password, you can also connect through the iw command, here take the MYIR_TECH hot spot as an example, the execution process is as follows:

```
root@mys-8mmx:~# wpa_cli -p/var/run/wpa_supplicant remove_network 0
Selected interface 'wlan0'
OK

root@mys-8mmx:~# wpa_cli -p/var/run/wpa_supplicant ap_scan 1
Selected interface 'wlan0'
OK

root@mys-8mmx:~# wpa_cli -p/var/run/wpa_supplicant add_network
Selected interface 'wlan0'
0

root@mys-8mmx:~# wpa_cli -p/var/run/wpa_supplicant set_network 0 ssid '"MYIR_TECH"'
```

```
Selected interface 'wlan0'
OK
root@mys-8mmx:~# wpa_cli -p/var/run/wpa_supplicant set_network 0 key_mgmt WPA-PSK
Selected interface 'wlan0'
OK
root@mys-8mmx:~# wpa_cli -p/var/run/wpa_supplicant set_network 0 psk '"myir@2016"'
Selected interface 'wlan0'
OK
root@mys-8mmx:~#  wpa_cli -p/var/run/wpa_supplicant select_network 0
Selected interface 'wlan0'
OK
```

- **Check Connection**

```
root@mys-8mmx:~# wpa_cli -p/var/run/wpa_supplicant status
Selected interface 'wlan0'
bssid=30:fc:68:9a:e8:99
freq=2437
ssid=MYIR_TECH
id=0
mode=station
wifi_generation=4
pairwise_cipher=CCMP
group_cipher=CCMP
key_mgmt=WPA2-PSK
wpa_state=COMPLETED
address=18:93:7f:74:5e:32
uuid=88ad20bf-14b0-55ce-bd17-faae5dd5d7c4
```

You also need /etc/wpa_supplicant.conf configuration file if you connect manually,here's how to generate this profile with wpa_passphrase.

```
root@mys-8mmx:~# head -n 4 /etc/wpa_supplicant.conf > /etc/wpa_supplicant.conf.tmp
```

```
root@mys-8mmx:~# wpa_passphrase MYIR_TECH myir@2016 >> /etc/wpa_supplicant.conf.tmp
root@mys-8mmx:~# mv /etc/wpa_supplicant.conf /etc/wpa_supplicant.conf.bak
root@mys-8mmx:~# mv /etc/wpa_supplicant.conf.tmp /etc/wpa_supplicant.conf
root@mys-8mmx:~# cat /etc/wpa_supplicant.conf
root@mys-8mmx:~# cat /etc/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
update_config=1

network={
        ssid="MYIR_TECH"
        #psk="myir@2016"
        psk=6a85bda077eebd8473c9b585b74c823199c6cbbe66a4f1d9e40c9b94e9605aa0
}
```

"MYIR_TECH" means wifi hot spot name ," myir@2016" is the wifi password,
Assign IP Address for Wi-Fi:

```
root@mys-8mmx:~# udhcpc -i wlan0
udhcpc: started, v1.31.0
udhcpc: sending discover
udhcpc: sending select for 192.168.40.150
udhcpc: lease of 192.168.40.150 obtained, lease time 7200
/etc/udhcpc.d/50default: Adding DNS 223.5.5.5
/etc/udhcpc.d/50default: Adding DNS 201.104.111.114
```

2) Connect to Wi-Fi Hotspots Automatically

The previous section described the manual connection of WiFi hotspots. It is very inconvenient to manually connect or manually execute scripts every time you start. Here we will introduce how to automatically connect WiFi hotspots when starting up. We organized the process of manually configuring WiFi into a script, Then set the boot to automatically execute this script, you can automatically generate */etc/wpa_supplicant.conf* configuration files and automatically enable wpa_supplicant service,here are steps to introduce how to automatically connect wifi.

- **Create wpa_supplicant.conf Profile**

Created by /usr/bin/ifup_wifi_sta the script file, this profile reads as follows:

```
       ...............
         rm ${WPA_FILE}
          echo "ctrl_interface=/var/run/wpa_supplicant" > ${WPA_FILE}
          echo "ctrl_interface_group=0" >> ${WPA_FILE}
          echo "update_config=1" >> ${WPA_FILE}
          echo "" >> ${WPA_FILE}
          wpa_passphrase ${SSID} ${PASSWD} >> ${WPA_FILE}
        .................
```

- **View wpa_supplicant Services**

```
root@mys-8mmx:~#  ls -l /lib/systemd/system/wpa_supplicant*
-rw-r--r-- 1 root root 453 Nov 18 08:30 /lib/systemd/system/wpa_supplicant-n
l80211@.service
-rw-r--r-- 1 root root 447 Nov 18 08:30 /lib/systemd/system/wpa_supplicant-w
ired@.service
-rw-r--r-- 1 root root 245 Nov 18 08:30 /lib/systemd/system/wpa_supplicant.s
ervice
-rw-r--r-- 1 root root 415 Nov 18 08:30 /lib/systemd/system/wpa_supplicant
@.service
```

- **Enabling wpa_supplicant services**

```
root@mys-8mmx:~# systemctl enable wpa_supplicant@wlan0.service
Created symlink /etc/systemd/system/multi-user.target.wants/wpa_supplicant@
wlan0.service -> /lib/systemd/system/wpa_supplicant@.service.
```

When the service is first started, you can see that a boot service wpa_supplicant@wlan0.service was created in the /etc/systemd/system/multi-user.target.wants/ directory, as follows:

```
root@mys-8mmx:~# cat /etc/systemd/system/multi-user.target.wants/wpa_suppl
icant@wlan0.service
```

```
[Unit]
Description=WPA supplicant daemon (interface-specific version)
Requires=sys-subsystem-net-devices-%i.device
After=sys-subsystem-net-devices-%i.device
Before=network.target
Wants=network.target

# NetworkManager users will probably want the dbus version instead.

[Service]
Type=simple
ExecStart=/usr/sbin/wpa_supplicant -c/etc/wpa_supplicant/wpa_supplicant-%I.conf -i%I

[Install]
WantedBy=multi-user.target
```

ExecStart parameter in the service represents a command or script to boot,so we can put the generated configuration files(wpa_supplicant.conf) in *\/etc/wpa_supplicant/* directory and then it starts automatically. Then we'll change the WPA_FILE in the ifup_wifi_sta script to */etc/wpa_supplicant/wpa_supplicnat-${ WLAN}.}-${WLAN}.conf*, If you want to know more about the script, please view the script source file(*/usr/bin/ifup_wifi_sta*).

3) Configured as a Hotspot for AP Mode Manually

Hostapd (Host access point daemon) is a user space software access point capable of turning normal network interface cards into access points and authentication servers. As a WiFi hotspot, the board needs to assign IP, routing and other network parameters for each terminal (such as mobile phone) that accesses the hotspot. For example, create SSID as MYIR_TEST and PASSWD as myir2020 wifi hotspot. Now let's do the configuration manually:

- **wlan on**

```
root@mys-8mmx:~# rfkill unblock wlan
```

- **Configure IP Address for wlan0**

```
root@mys-8mmx:~# ifconfig wlan0 up 192.168.10.1
```

As explained earlier, the current Wi-Fi module does not support STA and AP mode to work simultaneously, so the configuration of STA mode needs to be cleared here, as shown below:

```
root@mys-8mmx:~#  killall udhcpc
root@mys-8mmx:~#  killall wpa_supplicant
```

- **Run DHCP Server on wlan0**

“wlan0” works in AP mode. When other devices connect to this AP hotspot, it needs to dynamically assign IP addresses to other devices through the wireless connection, so it needs to use “wlan0” to run the DHCP service “udhcpd”. The configuration file corresponding to “udhcpd” is “/etc/udhcpd.conf”, which is as follows:

```
# the start and end of the IP lease block
start             192.168.10.10
end                192.168.10.254
# the interface that udhcpd will use
interface         wlan0

opt      dns      8.8.8.8
option   subnet   255.255.255.0
opt      router   192.168.10.1
option   domain   local
option   lease    864000
```

When AP mode configuration is completed and other devices are connected to this hotspot, the IP address in the above address pool will be obtained through “wlan0” with address range of 192.168.10.10~192.168.10.254, subnet mask of 255.255.255.0, default gateway of 192.168.10.1, DNS of 8.8.8.8, and DHCP lease time of 864000 seconds.

After the “udhcpd” configuration file is ready, execute the following command to start the “udhcpd” service with the following command:

```
root@mys-8mmx:~# udhcpd /etc/udhcpd.conf
```

- **Run hostapd on wlan0**

The most critical step in configuring the AP mode is of course to start the HostAPD service. Before starting the service, SSID, password, encryption algorithm, driver type, working mode, etc. of AP mode should be configured through */etc/hostapd.conf*. For the complete parameter configuration instructions, please refer to: http://w1.fi/cgit/hostap/plain/hostapd/hostapd.conf.

Here is the configuration of "/etc/hostapd.conf" for the current hardware, as follows:

```
# File: /etc/hostapd.conf
interface=wlan0
driver=nl80211
# mode Wi-Fi (a = IEEE 802.11a, b = IEEE 802.11b, g = IEEE 802.11g)
hw_mode=g
ssid=MYIR_TEST
channel=7
wmm_enabled=0
macaddr_acl=0
# Wi-Fi closed, need an authentication
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=myir2020
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

After the configuration file is ready, execute the following command to launch the HostAPD service, and the hot spots configured above can be used normally.

```
root@mys-8mmx:~# hostapd -B /etc/hostapd.conf
Configuration file: /etc/hostapd.conf
```

```
[  497.064719] [dhd-wlan0] wl_cfg80211_del_station : Disconnect STA : 66:66:3a:66:66:3a scb_val.val 3
Using interface wlan0 with hwaddr 18:93:7f:74:5e:32 and ssid "MYIR_TEST"
[  497.090180] [dhd-wlan0] wl_cfg80211_set_channel : netdev_ifidx(3), chan_type(1) target channel(7)
[  497.100044] [dhd] CFG80211-ERROR) wl_cfg80211_parse_ies : No WPSIE in beacon
[  497.107219] [dhd] CFG80211-ERROR) wl_cfg80211_parse_ies : No WPSIE in beacon
[  497.118244] [dhd] _dhd_wlfc_mac_entry_update():1866, entry(32)
[  497.126305] [dhd-wlan0] wl_cfg80211_bcn_bringup_ap : Creating AP with sec=wpa2psk/mfpn/0x44
[  497.192745] [dhd-wlan0] wl_iw_event : Link UP with 18:93:7f:74:5e:32
[  497.194112] [dhd-wlan0] wl_notify_connect_status_ap : AP/GO Link up
[  497.199141] [dhd-wlan0] wl_ext_iapsta_event : [A] Link up w/o creating? (etype=16)
[  497.213520] [dhd-wlan0] wl_iw_event : Link UP with 18:93:7f:74:5e:32
[  497.219973] [dhd-wlan0] wl_ext_iapsta_event : [A] Link up w/o creating? (etype=16)
[  497.233527] [dhd] CFG80211-ERROR) wl_set_wsec_info_algos : wsec_info error (-23)
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

- **Wifi Connection**

Using the mobile phone to connect "MYIR_TEST" wifi hot spots, you can see the automatically obtained IP address(192.168.10.10) on the mobile phone, but the phone can not connect to the external network.

```
root@mys-8mmx:~# ping 192.168.10.10
PING 192.168.10.10 (192.168.10.10) 56(84) bytes of data.
64 bytes from 192.168.10.10: icmp_seq=1 ttl=64 time=14.7 ms
64 bytes from 192.168.10.10: icmp_seq=2 ttl=64 time=173 ms
64 bytes from 192.168.10.10: icmp_seq=3 ttl=64 time=16.1 ms
64 bytes from 192.168.10.10: icmp_seq=4 ttl=64 time=13.4 ms
```

- **Setting up Network Forwarding**

If the Ethernet card eth0 is already connected to the Internet, then IP forwarding is done through the following configuration, and the devices connected to “MYIR_TEST” can also be connected to the Internet.

```
root@mys-8mmx:~# echo "1" > /proc/sys/net/ipv4/ip_forward
root@mys-8mmx:~# iptables -t nat -A POSTROUTING -s 192.168.10.0/24 -o e
th0 -j MASQUERADE
[  641.528883] audit: type=1325 audit(1608196314.812:3): table=nat family=2
entries=0
[  641.536548] audit: type=1325 audit(1608196314.812:4): table=nat family=2
entries=5
```

4) Configure Wi-Fi AP Mode by a SHELL Script

We compiled the manual Wi-Fi configuration process into a script named as “ifup_wifi_ap” and provided it to the user for reference. The user only needs to prepare “*/etc/udhcpd.conf*” and “*/etc/hostapd.conf*” configuration files, and then execute this script to configure the AP mode. The script contents are as follows：

```
#!/usr/bin/env sh
# File: /usr/bin/ifup_wifi_ap.sh
ETH=eth0
WLAN=wlan0
WLAN_IP=192.168.10.1
DHCP_FILE=/etc/udhcpd.conf
HOSTAPD_FILE=/etc/hostapd.conf

clean_stage(){
    killall udhcpc
    killall wpa_supplicant

    killall hostapd
    killall udhcpd
    sleep 1
}

enable_wifi(){
    T_HCI="phy0"
    RFKILL_SYS_PATH="/sys/class/rfkill/"
    dir=`ls ${RFKILL_SYS_PATH}`
    for i in ${dir}
    do
      if [ ${T_HCI} == `cat ${RFKILL_SYS_PATH}${i}/name` ];then
          echo 0 > ${RFKILL_SYS_PATH}${i}/state
          echo "find ${T_HCI} enable it"
          sleep 1
          echo 1 > ${RFKILL_SYS_PATH}${i}/state
      fi
    done
 }

enable_ap_mode(){
    ifconfig ${WLAN} up ${WLAN_IP}

    echo "1" > /proc/sys/net/ipv4/ip_forward
    iptables -t nat -A POSTROUTING -s ${WLAN_IP}/24 -o ${ETH} -j MASQUERADE

    sleep 1
    udhcpd ${DHCP_FILE}
    hostapd -B ${HOSTAPD_FILE}
}

clean_stage
enable_wifi
enable_ap_mode
```

The AP mode can be configured by executing “*/usr/bin/ifup_wifi_ap*” under the Linux system console. After successful execution, the user can use another device to connect to the MYIR_TEST hot spot for testing.

4.3. Bluetooth

BlueZ (http://www.bluez.org/) is usually used for the configuration and management of Bluetooth devices under the Linux platform. BlueZ is a relatively complete set of Blutooth configuration and management tools and protocol stack. The following tools are used to configure and use Bluetooth.

The WIFI/Bluetooth Combo Module(AP6212) is tested here, and its corresponding drivers are "hci_uart" module and "btbcm" module. The results are shown below through the "lsmod" command:

```
root@mys-8mmx:~#  lsmod
..................
bcmdhd                    1286144   0
snd_soc_imx_wm8904          16384   0
ov5640_camera_mipi_v2       24576   0
```

The "hci0" device node will be generated after successful driver loading. The following details describe the process of configuring and connecting to a nearby Bluetooth device using the BlueZ toolkit:

1) Binding Port

Execute the following command, which automatically opens bluetoothd service:

```
 root@mys-8mmx:~# brcm_patchram_plus -d --enable_hci --no2bytes --toslee
p 200000 --baudrate 3000000 --patchram /lib/firmware/bcmd/bcm43438a1.hcd
 /dev/ttymxc0 &
```

2) Activate hci0

If the switch of Bluetooth is not turned on, you can check the current switching state of Bluetooth through RFKILL and make corresponding processing.

```
root@mys-8mmx:~# rfkill unblock bluetooth
root@mys-8mmx:~# hciconfig hci0 up
```

3) Scan the Bluetooth Devices Nearby

```
root@mys-8mmx:~# hcitool scan
```

```
Scanning ...
        74:0A:E1:35:D0:FA       Metro
```

4) Manage Bluetooth by bluetoothctl

Bluetoothctl is a set of Bluetooth management tools from BlueZ that enables Bluetooth controller power, proxies, scanning, matching, and connection with this command. Before you use bluetoothctl verification system of bluetooth service is started, or confirm the */usr/libexec/bluetooth/bluetoothd* service program is started.

```
root@mys-8mmx:~# systemctl status bluetooth.service
* bluetooth.service - Bluetooth service
   Loaded: loaded (/lib/systemd/system/bluetooth.service; enabled; vendor pre
set: enabled)
   Active: active (running) since Thu 2020-12-17 09:15:09 UTC; 3min 4s ago
     Docs: man:bluetoothd(8)
 Main PID: 742 (bluetoothd)
   Status: "Running"
    Tasks: 1
   Memory: 2.1M
   CGroup: /system.slice/bluetooth.service
           `-742 /usr/libexec/bluetooth/bluetoothd

Dec 17 09:15:09 myd-imx8mm systemd[1]: Starting Bluetooth service...
Dec 17 09:15:09 myd-imx8mm systemd[1]: Started Bluetooth service.
```

Below are the steps for using Bluetoothctl to test Bluetooth specifically, pairing, and connecting：

- **Run bluetoothctl to Enter the Bluetooth Control Interface**

```
root@mys-8mmx:~#  bluetoothctl
Agent registered
[bluetooth]#
```

- **Power on Bluetooth**

```
[bluetooth]# power on
```

```
Changing power on succeeded
```

- **Enable Bluetooth Agent**

Manage and check whether the Bluetooth agent is successful registered, which is registered by default.

```
[bluetooth]# agent on
Agent is already registered
[bluetooth]# default-agent
Default agent request successfu
```

- **Scan the Bluetooth Devices Nearby**

```
[bluetooth]# scan on
Discovery started
[CHG] Controller 18:93:7F:74:5E:33 Discovering: yes
[NEW] Device 5A:5A:74:AF:02:11 5A-5A-74-AF-02-11
[NEW] Device 63:D5:E7:37:20:64 63-D5-E7-37-20-64
[NEW] Device D4:61:9D:1F:76:35 D4-61-9D-1F-76-35
[CHG] Device 5A:5A:74:AF:02:11 RSSI: -75
[CHG] Device 5A:5A:74:AF:02:11 RSSI: -86
[CHG] Device 5A:5A:74:AF:02:11 RSSI: -73
[CHG] Device 5A:5A:74:AF:02:11 RSSI: -87
[NEW] Device 74:0A:E1:35:D0:FA Metro
```

It can be seen from the above that a mobile phone of Mi with MAC address of 74:0A:E1:35:D0:FA Metro has been scanned. The following is the pairing and connecting test with this device.

- **Pair the Bluetooth Device**

```
[bluetooth]# pair 74:0A:E1:35:D0:FA
Attempting to pair with 74:0A:E1:35:D0:FA
[CHG] Device 74:0A:E1:35:D0:FA Connected: yes
Request confirmation
[agent] Confirm passkey 910624 (yes/no): yes
[CHG] Device 74:0A:E1:35:D0:FA Modalias: bluetooth:v010Fp107Ed1436
```

```
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 0000046a-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 0000fdd1-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA UUIDs: 0000fe35-0000-1000-8000-00805f9b34fb
[CHG] Device 74:0A:E1:35:D0:FA ServicesResolved: yes
[CHG] Device 74:0A:E1:35:D0:FA Paired: yes
Pairing successful
```

```
[CHG] Device 74:0A:E1:35:D0:FA ServicesResolved: no
[CHG] Device 74:0A:E1:35:D0:FA Connected: no
[bluetooth]#
```

So far, the device has been paired successfully with HUAWEI mobile phone Bluetooth.

- **Connect Bluetooth Device**

```
[bluetooth]# connect 74:0A:E1:35:D0:FA
Attempting to connect to 74:0A:E1:35:D0:FA
[CHG] Device 74:0A:E1:35:D0:FA Connected: yes
Connection successful
[CHG] Device 74:0A:E1:35:D0:FA ServicesResolved: yes
[Metro]#
```

So far, you will see Bluetooth and connection on the phone.

4.4. 4G/5G

LINUX device can also dial through an external 4 G or 5 modules,of these ,4 G modules use more EC20,5G modules include rm500q and RG801H.

There are pppd,gobinet and qmi_wwan 3 ways to dial, in which the pppd is more common, the gobinet is not used, the qmi_wwan connection is faster, and can be used to do 5 module connections.

Take ppp and qmi_wwan dialing as an example, using USB transfer board and EC20CE FDKG module to test.

1) Check VID and PID

Install the EC20 module on the USB transfer board, connect to the development board, and then use the lsusb to view the EC20 module information.

```
root@mys-8mmx:~# lsusb
Bus 001 Device 005: ID 2c7c:0125 Quectel Wireless Solutions Co., Ltd. EC25 LTE modem
Bus 001 Device 003: ID 0951:1666 Kingston Technology DataTraveler 100 G3/G4/SE9 G2
Bus 001 Device 002: ID 0424:2514 Standard Microsystems Corp. USB 2.0 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

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

Then you need to configure the following in the array *"static const struct usb_device_id option_ids"* in *${KERNEL_DIR}/drivers/usb/serial/option.c:*

```
#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--
}
```

kernel need to open these configurations:

```
+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 kernel Identification Module

The nodes are generated /dev/ttyUSB* if the VID and PID configuration of this module is added to the kernel source code:

```
root@mys-8mmx:~# ls -l /dev/ttyUSB*
crw-rw---- 1 root dialout 188, 0 Dec  9 10:18 /dev/ttyUSB0
crw-rw---- 1 root dialout 188, 1 Dec  9 10:18 /dev/ttyUSB1
crw-rw---- 1 root dialout 188, 2 Dec  9 10:18 /dev/ttyUSB2
crw-rw---- 1 root dialout 188, 3 Dec  9 10:18 /dev/ttyUSB3
```

3) Initial testing using AT Instructions

Take the microcom tool as an example, or minicom. Such as: execute microcom /dev/ttyUSB2 command to enter test mode, ctrl x exit.

- **Detection of Signal Quality**

```
root@mys-8mmx:~#  microcom /dev/ttyUSB2
at+csq
+CSQ: 22,99

OK
```

- ➢ 22,99: 22：Signal quality, the smaller the number, the stronger the signal.

- **View Ready Status**

```
at+cpin?
+CPIN: READY

OK
```

- ➢ +CPIN:READY : READY Indicates readiness.

- **View Operators**

```

at+cops?
+COPS: 0,0,"CHN-UNICOM",2

OK
```

- ➢ "CHN-UNICOM",2: CHN-UNICOM Representing Unicom,

If the above 3 steps can be normal, you can dial and connect to the network, the following will also describe how to call and text messages.

- **Call Someone**

```
ATD134xxxx5673;
OK
```

- **Send Message**

```
at+cmgf=1
OK
at+cscs="GSM"
OK
at+cmgs="134xxxx5673"
> hello[   63.568240] audit: type=1006 audit(1607509801.160:3): pid=741 uid=
0 old-auid=4294967295 auid=0 tty=(none) old-ses=4294967295 ses=2 res=1

+CMGS: 76
OK
```

- ➢ At+cmgf=1: Set Text Message Mode
- ➢ At+cscs=" GSM" : Set TE to use GSM characters
- ➢ At+cmgs: "Phone Number" 'CTRL+ Z' sends the message, 'ECS' button exit message sent

4) ppp Dial Test

Dial commands with pppd development board:

```
root@mys-8mmx:~# pppd call quectel-dial
root@mys-8mmx:~# [    33.774572] [dhd] CFG80211-ERROR) wl_cfg80211_netd
ev_notifier_call : wdev null. Do nothing
[    33.783789] [dhd] CFG80211-ERROR) wl_cfg80211_netdev_notifier_call : wde
v null. Do nothing
[    37.259047] [dhd] CFG80211-ERROR) wl_cfg80211_netdev_notifier_call : wde
v null. Do nothing
[    37.267814] [dhd] CFG80211-ERROR) wl_cfg80211_netdev_notifier_call : wde
v null. Do nothing
```

Dial needs to wait a while, dial log has been hidden, users can view the corresponding log:

```
root@mys-8mmx:~#  cat /var/log/quectel-dial.log
pppd options in effect:
debug           # (from /etc/ppp/peers/quectel-dial)
persist         # (from /etc/ppp/peers/quectel-dial)
logfile /var/log/quectel-dial.log             # (from /etc/ppp/peers/quectel-d
ial)
maxfail 0               # (from /etc/ppp/peers/quectel-dial)
dump            # (from /etc/ppp/peers/quectel-dial)
noauth          # (from /etc/ppp/peers/quectel-dial)
user card               # (from /etc/ppp/peers/quectel-dial)
password ??????         # (from /etc/ppp/peers/quectel-dial)
remotename 3gppp                # (from /etc/ppp/peers/quectel-dial)
/dev/ttyUSB3            # (from /etc/ppp/peers/quectel-dial)
115200          # (from /etc/ppp/peers/quectel-dial)
lock            # (from /etc/ppp/peers/quectel-dial)
connect chat -s -v -f /etc/ppp/chatscripts/quectel-chat-connect         # (fro
m /etc/ppp/peers/quectel-dial)
disconnect chat -s -v -f /etc/ppp/chatscripts/quectel-chat-disconnect
 # (from /etc/ppp/peers/quectel-dial)
```

```
nocrtscts                 # (from /etc/ppp/peers/quectel-dial)
local          # (from /etc/ppp/peers/quectel-dial)
lcp-echo-failure 12                 # (from /etc/ppp/peers/quectel-dial)
lcp-echo-interval 5                 # (from /etc/ppp/peers/quectel-dial)
hide-password               # (from /etc/ppp/peers/quectel-dial)
ipcp-accept-local                   # (from /etc/ppp/peers/quectel-dial)
ipcp-accept-remote                   # (from /etc/ppp/peers/quectel-dial)
ipparam 3gppp              # (from /etc/ppp/peers/quectel-dial)
noipdefault               # (from /etc/ppp/peers/quectel-dial)
ipcp-max-failure 10                 # (from /etc/ppp/peers/quectel-dial)
defaultroute              # (from /etc/ppp/peers/quectel-dial)
usepeerdns                # (from /etc/ppp/peers/quectel-dial)
noccp           # (from /etc/ppp/peers/quectel-dial)
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 (AT+CGATT=0^M)
```

```
expect (OK)
^M
^M
OK
 -- got it

send (AT+CGATT=1^M)
expect (OK)
^M
^M
OK
 -- got it

send (ATI;+CSUB;+CSQ;+CPIN?;+COPS?;+CGREG?;&D2^M)
expect (OK)
^M
^M
Quectel^M
EC20F^M
Revision: EC20CEFDKGR06A07M2G^M
^M
SubEdition: V03^M
^M
+CSQ: 22,99^M
^M
+CPIN: READY^M
^M
+COPS: 0^M
^M
+CGREG: 0,1^M
^M
OK
 -- got it
```

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

send (ATDT*99#^M)
expect (CONNECT)
^M
^M
CONNECT
 -- got it

Script chat -s -v -f /etc/ppp/chatscripts/quectel-chat-connect finished (pid 74
0), status = 0x0
Serial connection established.
using channel 1
Using interface ppp0
Connect: ppp0 <--> /dev/ttyUSB3
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xbb05c284> <pcomp>
<accomp>]
rcvd [LCP ConfReq id=0x0 <asyncmap 0x0> <auth pap> <magic 0x8527ca8
4> <pcomp> <accomp>]
sent [LCP ConfAck id=0x0 <asyncmap 0x0> <auth pap> <magic 0x8527ca8
4> <pcomp> <accomp>]
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xbb05c284> <pcomp>
<accomp>]
sent [LCP EchoReq id=0x0 magic=0xbb05c284]
sent [PAP AuthReq id=0x1 user="card" password=<hidden>]
rcvd [LCP DiscReq id=0x1 magic=0x8527ca84]
rcvd [LCP EchoRep id=0x0 magic=0x8527ca84 bb 05 c2 84]
rcvd [PAP AuthAck id=0x1 ""]
PAP authentication succeeded
sent [IPCP ConfReq id=0x1 <compress VJ 0f 01> <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfNak id=0x1 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
sent [IPCP ConfReq id=0x2 <compress VJ 0f 01> <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
rcvd [IPCP ConfNak id=0x2 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
sent [IPCP ConfReq id=0x3 <compress VJ 0f 01> <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]
rcvd [IPCP ConfRej id=0x3 <compress VJ 0f 01> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]
sent [IPCP ConfReq id=0x4 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfNak id=0x4 <addr 10.241.79.143> <ms-dns1 218.104.111.114> <ms-dns2 218.104.111.122>]
sent [IPCP ConfReq id=0x5 <addr 10.241.79.143> <ms-dns1 218.104.111.114> <ms-dns2 218.104.111.122>]
rcvd [IPCP ConfAck id=0x5 <addr 10.241.79.143> <ms-dns1 218.104.111.114> <ms-dns2 218.104.111.122>]
Could not determine remote IP address: defaulting to 10.64.64.64
not replacing default route to eth0 [192.168.40.1]
local  IP address 10.241.79.143
remote IP address 10.64.64.64
primary   DNS address 218.104.111.114
```

```
secondary DNS address 218.104.111.122
Script /etc/ppp/ip-up started (pid 749)
Script /etc/ppp/ip-up finished (pid 749), status = 0x0
```

From above you can see that the connection is normal and can get the IP normally.

Execute the following command to view "ppp0". It is found that IP has been obtained normally and can ping Baidu normally:

```
root@mys-8mmx:~# ifconfig ppp0
ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.241.79.143  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:
1
          RX packets:7 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:130 (130.0 B)  TX bytes:414 (414.0 B)

root@mys-8mmx:~# ping www.baidu.com
PING www.baidu.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=54 time=19.6 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=19.4 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=19.5 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=19.6 ms
```

- ppp0 ： Dial-up network card, normal access to the ip address.

5) qmi_wwan Dial Test

Use quectel-CM command to dial,eg: quectel-CM -s 3gnet &

```
root@mys-8mmx:~# quectel-CM -s 3gnet &
[1] 740
[12-09_10:51:10:212] WCDMA&LTE_QConnectManager_Linux&Android_V1.1.34
[12-09_10:51:10:212] quectel-CM profile[1] = 3gnet///0, pincode = (null)
[12-09_10:51:10:213] Find /sys/bus/usb/devices/1-1.3 idVendor=2c7c idProduct
=0125
[12-09_10:51:10:213] Find /sys/bus/usb/devices/1-1.3:1.4/net/wwan0
```

```
[12-09_10:51:10:213] Find usbnet_adapter = wwan0
[12-09_10:51:10:213] Find /sys/bus/usb/devices/1-1.3:1.4/usbmisc/cdc-wdm0
[12-09_10:51:10:213] Find qmichannel = /dev/cdc-wdm0
[12-09_10:51:10:228] cdc_wdm_fd = 7
root@mys-8mmx:~# [12-09_10:51:10:311] Get clientWDS = 18
[12-09_10:51:10:343] Get clientDMS = 1
[12-09_10:51:10:375] Get clientNAS = 3
[12-09_10:51:10:407] Get clientUIM = 1
[12-09_10:51:10:439] Get clientWDA = 1
[12-09_10:51:10:469] requestBaseBandVersion EC20CEFDKGR06A07M2G
[12-09_10:51:10:567] requestGetSIMStatus SIMStatus: SIM_READY
[12-09_10:51:10:567] requestSetProfile[1] 3gnet///0
[12-09_10:51:10:630] requestGetProfile[1] 3gnet///0
[12-09_10:51:10:663] requestRegistrationState2 MCC: 460, MNC: 1, PS: Attache
d, DataCap: UMTS
[12-09_10:51:10:694] requestQueryDataCall IPv4ConnectionStatus: DISCONNECT
ED
[12-09_10:51:10:758] requestRegistrationState2 MCC: 460, MNC: 1, PS: Attache
d, DataCap: UMTS
[12-09_10:51:13:255] requestSetupDataCall WdsConnectionIPv4Handle: 0xe1769
c90
[12-09_10:51:13:319] requestRegistrationState2 MCC: 460, MNC: 1, PS: Attache
d, DataCap: UMTS
[12-09_10:51:13:351] requestQueryDataCall IPv4ConnectionStatus: CONNECTED
[12-09_10:51:13:415] ifconfig wwan0 up
[   42.787033] [dhd] CFG80211-ERROR) wl_cfg80211_netdev_notifier_call : wde
v null. Do nothing
[   42.796362] [dhd] CFG80211-ERROR) wl_cfg80211_netdev_notifier_call : wde
v null. Do nothing
[12-09_10:51:13:446] busybox udhcpc -f -n -q -t 5 -i wwan0
udhcpc: started, v1.31.0
udhcpc: sending discover
udhcpc: sending select for 10.42.46.46
```

```
udhcpc: lease of 10.42.46.46 obtained, lease time 7200
[12-09_10:51:13:640] /etc/udhcpc.d/50default: Adding DNS 218.106.127.114
[12-09_10:51:13:640] /etc/udhcpc.d/50default: Adding DNS 218.104.111.122
[12-09_10:51:20:326] requestRegistrationState2 MCC: 460, MNC: 1, PS: Attache
d, DataCap: UMTS
```

- -s ctnet chinatelecom CTCC
- -s cmnet chinamobile CMCC
- -s 3gnet chinaunicom CUCC

Execute ifconfig commands to view network information:

```
root@mys-8mmx:~# ifconfig
eth0      Link encap:Ethernet  HWaddr 36:2a:6e:3e:42:f2
          inet addr:192.168.40.202  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::342a:6eff:fe3e:42f2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1036 errors:0 dropped:0 overruns:0 frame:0
          TX packets:127 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:97384 (95.1 KiB)  TX bytes:14852 (14.5 KiB)

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:82 errors:0 dropped:0 overruns:0 frame:0
          TX packets:82 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6220 (6.0 KiB)  TX bytes:6220 (6.0 KiB)

wwan0     Link encap:Ethernet  HWaddr 0e:84:fc:87:fd:cb
          inet addr:10.42.46.46  Bcast:10.42.46.47  Mask:255.255.255.252
          UP BROADCAST RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

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

```
root@mys-8mmx:~# ping www.baidu.com
PING www.baidu.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=54 time=20.1 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=2 ttl=54 time=19.7 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=3 ttl=54 time=19.5 ms
64 bytes from 14.215.177.39 (14.215.177.39): icmp_seq=4 ttl=54 time=19.4 ms
```

5. Network Applications

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

5.1. PING

PING is used primarily to test network connectivity. It can also test network latency and packet loss rates. Once the Ethernet connection is configured as in 4.1, PING can be used for simple testing of network connections.

1) Network Connection

When you connect a device to a switch or router via a CAT6 cable, the console displays the connection information that the kernel outputs, as follows：

```
root@mys-8mmx:~# [ 4331.251965] fec 30be0000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx
```

2) PING Public Network

```
root@mys-8mmx:~# ping www.baidu.com -I eth0
PING www.baidu.com (14.215.177.38) from 192.168.40.202 eth0: 56(84) bytes of data.
64 bytes from 14.215.177.38: icmp_seq=1 ttl=54 time=19.5 ms
64 bytes from 14.215.177.38: icmp_seq=2 ttl=54 time=19.3 ms
64 bytes from 14.215.177.38: icmp_seq=3 ttl=54 time=19.3 ms
64 bytes from 14.215.177.38: icmp_seq=4 ttl=54 time=19.3 ms
64 bytes from 14.215.177.38: icmp_seq=5 ttl=54 time=19.3 ms
64 bytes from 14.215.177.38: icmp_seq=6 ttl=54 time=19.3 ms
64 bytes from 14.215.177.38: icmp_seq=7 ttl=54 time=19.3 ms
```

NOTE：Ping public network needs to ensure that DNS is working properly.

The above results show that the IP address of “www.baidu.com” after domain name resolution is 112.80.248.75, ICMP_SEQ stands for the number of ICMP packet, if the number is serial, no packet is lost; Time stands for the latency of the response, and the shorter the better, of course. In addition to testing Ethernet, the ping command can also be used to test Wi-Fi.

5.2. SSH

SSH stands for Secure Shell and was developed by the IETF's Network Working Group. SSH is an application layer based security protocol, which is more reliable and provides security for remote login sessions and other network services. Typically, Linux platforms use Dropbear or OpenSSH to implement both the server and client sides of SSH. Let's test the use of SSH client and server on Ethernet connection respectively. Current factory default includes OpenSSH 7.6 P1 (http://www.openssh.com/) client and server.

The Ethernet interface connection to SSH server is configured in accordance with Section 4.1. The configured Ethernet card address is as follows:

```
root@mys-8mmx:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr ee:2b:15:2d:1c:12
          inet addr:192.168.40.202  Bcast:192.168.40.255  Mask:255.255.255.0
          inet6 addr: fe80::ec2b:15ff:fe2d:1c12/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:44817 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22097 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:27374893 (26.1 MiB)  TX bytes:3013121 (2.8 MiB)
```

The SSH server's IP address is 192.168.40.2. The following test can be conducted after the connection between the test device and SSH server is normal with the ping command.

```
root@mys-8mmx:~# ping 192.168.40.2
PING 192.168.40.2 (192.168.40.2) 56(84) bytes of data.
64 bytes from 192.168.40.2: icmp_seq=1 ttl=64 time=2.45 ms
64 bytes from 192.168.40.2: icmp_seq=2 ttl=64 time=1.18 ms
64 bytes from 192.168.40.2: icmp_seq=3 ttl=64 time=1.16 ms
64 bytes from 192.168.40.2: icmp_seq=4 ttl=64 time=1.83 ms
64 bytes from 192.168.40.2: icmp_seq=5 ttl=64 time=1.16 ms
```

- **SSH Client Test**

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

```
root@mys-8mmx:~# ssh myir@192.168.40.2

Host '192.168.40.2' is not in the trusted hosts file.
(ecdsa-sha2-nistp256 fingerprint sha1!! 17:59:83:29:04:cc:97:63:b8:82:52:73:3c:47:
70:80:8a:5d:2a:37)
Do you want to continue connecting? (y/n) y
myir@192.168.40.2's password:
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-101-generic x86_64)

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

779 scalable packages.
573 security updates

New release '18.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Dec 17 08:47:52 2020 from 192.168.40.88
myir@myir-server1:~$
```

After successful login, automatically enter the console on the SSH server, and the user can perform their Linux user permission control on the remote server on the client side. If exit is required, simply execute the "exit" command from the current console.

- **SSH Server Test**

The device acts as an SSH server, and other devices are connected to the device remotely.

Since SSH service is also started on the device side by default, we can also use SSH commands on other devices with SSH clients to log in to the current device, with the following commands and results：

```
myir@myir-server1:~$ ssh root@192.168.40.202
The authenticity of host '192.168.40.100 (192.168.40.100)' can't be established.
RSA key fingerprint is SHA256:6T7Bvwtc+MU8nlbdoU9YH0qonwuvWPdb8D+F+
ae5ktA.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.40.202' (RSA) to the list of known host
s.
```

In the example above, we remotely log on to the device as root and go to the console to perform root user privileges on the device. If you need to exit, simply execute the "exit" command from the console.

OpenSSH is the primary connection tool for remote login using the SSH protocol. It encrypts all traffic to eliminate eavesdropping, connection hijacking, and other attacks. In addition, OpenSSH provides a range of large secure tunneling capabilities, multiple authentication methods, and complex and flexible configuration options. Users can modify the configuration files “ssh_config” and “sshd_config” in the “*/etc/ssh/*” directory as they wish.

For example, if you want the SSH server to allow the root account to log in remotely without a password, you can modify “*/etc/ssh/sshd_config*” on the SSH server device to add the following two lines of configuration.

```
PermitRootLogin yes
PermitEmptyPasswords yes
```

The above configuration poses a significant security risk and is typically used for remote deployment during the debug phase. In real products, it's usually turned off for security reasons.

5.3. SCP

SCP, short for Secure Copy, is a Secure remote file Copy command based on the SSH protocol on Linux systems and is very useful during system debugging.
In 4.2, we have introduced the example of using SSH protocol and SSH client and server for remote login. Here, we introduce the example of remote copy of files through SCP command:

1) Copy Files from Remote to Local

```
myir@myir-server1:~/MYS-8MMX$ scp gpiotest root@192.168.40.100:/home/ro
ot
gpiotest
 100%    13KB   13.1KB/s    00:00
```

Go to the development board home directory to see this file, as shown below：

```
root@mys-8mmx:~# ls -l README-IMXBSP
-rwxr-xr-x 1 root root 4453 Dec 17 10:30 README-IMXBSP
root@mys-8mmx:~# pwd
/home/root
```

2) Copy Files form Local to Remote

```
root@mys-8mmx:~# scp README-IMXBSP myir@192.168.40.2:~/
myir@192.168.40.2's password:
README-IMXBSP                                   100% 4453       4.4KB/s     00:00
```

Go to server’ s current directory to see this file,as shown below:

```
root@mys-8mmx:~# ls -l gpiotest
-rwxr-xr-x 1 root root 13448 Nov 11 05:59 gpiotest
root@mys-8mmx:~# pwd
/home/root
```

During the copying process, please follow the prompts to input. After successful verification, the file is copied from the device to the $HOME directory of the specified account on the server.

You can also copy directories by adding the "-r" parameter, as the SCP command helps.

5.4. FTP

FTP(File Transfer Protocol) is a network protocol for file transfer. The protocol follows the client-server communication model. To transfer files using FTP, the user needs to run an FTP client program and start a connection to a remote computer running FTP server software. After establishing the connection, the client can choose to send and/or receive a copy of the file. The FTP server listens for connection requests from the FTP client on TCP port 21. When a request is received, the server USES this port to control the connection and opens a separate port to transfer file data.

The default factory myir-image-full image contains a FTP client program FTP and server-side proftpd (see section 7.3 in “MYS-8MMX Software Development Guide" for migration of FTP application). Start the development board, on target side the root directory of the FTP is at the “/var/lib/ftp”, the following tests within the local area network (LAN) using FTP commands anonymously to the target device and from the development of the host (IP address 192.168.40.88) to the target device (IP address for 192.168.40.202) transfer file example.

1) Login from Host PC to Target Device with FTP

The development host is a Windows10 system PC, open the command window, login to the development board with FTP, user name is “ftp”, password is arbitrary.

```
E:\ftp>ftp 192.168.40.202
connect 192.168.40.202。
220 ProFTPD Server (ProFTPD Default Installation) [192.168.40.202]
500 OPTS UTF8 not understood
用户(192.168.40.202:(none)): ftp
331 Anonymous login ok, send your complete email address as your passwor
d
密码:
230 Anonymous access granted, restrictions apply
ftp> pwd
```

```
257 "/" is the current directory
```

The current windows FTP working directory is the *E:\ftp*, development board working directory is */var/lib/ftp*, and then you can download and upload files.

2) Create Files on Target Device for Test

```
root@mys-8mmx:/var/lib/ftp# touch test
root@mys-8mmx:/var/lib/ftp# ls
test
```

3) View Files on Target Device with FTP at Host PC

```
ftp> ls
200 PORT command successful
150 Opening ASCII mode data connection for file list
test
226 Transfer complete
ftp: 收到 9 字节，用时 0.00 秒 2.25 千字节/秒。
```

4) Download Files from the Target Device with FTP

The FTP command line supports downloading multiple files using the “mget” subcommand, but each file requires user confirmation, as shown below:

```
ftp> mget aaa test
200 Type set to A
mget aaa? y
200 PORT command successful
150 Opening ASCII mode data connection for aaa
226 Transfer complete
mget test? y
200 PORT command successful
150 Opening ASCII mode data connection for test
226 Transfer complete
ftp> bye
221 Goodbye.
```

Execution command "bye" to exit the ftp service, and then the files downloaded from the development board can be found in the windows10 host disk E:\ftp directory, as follows:

```
E:\ftp>dir
 .........
2020/12/17   11:36    <DIR>          .
2020/12/17   11:36    <DIR>          ..
2020/12/17   11:36                 0  aaa
2020/12/17   11:36                 0  test
 ...........
```

5) Upload Files to the Target Device with FTP

FTP need ordinary user or root user login to upload files, here use root user login, user name is "root", no password, directly press" Enter" key, if you need password users can directly set the root password in the target development board:

- **Settings root User Password**

On the development board, enter the following command to set root password, where I set root password to "1":

```
root@mys-8mmx:~# passwd root
New password:
Retype new password:
passwd: password updated successfully
```

- **Modify Configuration and Restart Service**

To log in to the FTP server using a root account, you need to modify /etc/proftpd.conf the file and add a line of configuration "RootLogin on" to the file，and then restart the proftpd service:

```
systemctl restart proftpd
```

- **Connect target board with FTP**

```
E:\ftp>ftp 192.168.40.202
connect 192.168.40.202。
220 ProFTPD Server (ProFTPD Default Installation) [192.168.40.202]
```

```
500 OPTS UTF8 not understood
user(192.168.40.202:(none)): root
331 Password required for root
password:
230 User root logged in
ftp> pwd
257 "/home/root" is the current directory
```

- **Upload files**

FTP file transfer supports ASCII mode and binary mode. The default download file is ASCII mode. If it is to transfer binary files, such as executable programs that need to be executed on the target device, try to use binary mode. As follows:

```
ftp> binary
200 Type set to I
ftp> put mxapp2
200 PORT command successful
150 Opening BINARY mode data connection for mxapp2
226 Transfer complete
ftp: Sending 28259072 bytes takes 3.56 seconds and 7944.64 Kbytes/s.
ftp> bye
221 Goodbye.
```

FTP dates back to the early 1970s and was written with no security concerns. It does not use encryption on anything. The login credentials (such as username and password) and the data you download or upload are transmitted in clear text. Therefore, it is not suitable for transmitting sensitive information over the Internet. However, considering its good compatibility and stability, it is very convenient to use in LAN.

5.5. TFTP

Like FTP, TFTP uses client and server software to connect and transfer files between two devices, but the difference is that TFTP uses UDP protocol, which does not have the login function. It is very simple, especially suitable for transferring and backing up firmware, configuration files and other information on the device and server side. For example, TFTP protocol is supported in the common U-boot, which can load the server-side Linux system through the network and realize the function of network startup.

The default image file contains the TFTP client program provided by BusyBox, with the following command syntax:

```
root@mys-8mmx:~# tftp --help
BusyBox v1.31.0 (2020-11-18 08:19:02 UTC) multi-call binary.

Usage: tftp [OPTIONS] HOST [PORT]
```

The detailed parameters are described below:

- -g : Get
- -p : Upload/Put
- -l : Local files
- -r : Remote files
- HOST: Remote host name or IP address
- [PORT]: Optional remote host port, default is 69

Users can choose tftp-hpa on Linux, can also choose tftpd32/64 (http://tftpd32.jounin.net/tftpd32_download.html) on Windows as the TFTP server application. Now, we choose "tftpd-hpa" on Ubuntu as an example to show the TFTP server configuration.

1) Install TFTP Server Application

Install the TFTP server application on the Ubuntu server as follows:

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

2) Configure TFTP Server

Create the TFTP server working directory and open the TFTP service configuration file as follows, please replace <WORKDIR> with yourselves work directory:

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

Modify or add the following fields：

```
TFTP_DIRECTORY="<WORKDIR>/tftpboot"
TFTP_OPTIONS="-l -c -s"
```

3) Restart TFTP Service

Restart the TFTP service on the Ubuntu server as follows:

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

Once the TFTP server is configured, place a test file “*zImage*” in the configured “*<WORKDIR>/tftpboot/*” directory, and you can download and upload files using the TFTP client on the target device.

```
root@mys-8mmx:~# tftp -g -r uImage -l uImage 192.168.40.143
```

The above command downloads the “*zImage*” from the TFTP server “*<WORKDIR>/tftpboot/*” directory to the current directory of the target device.

```
root@mys-8mmx:~# tftp -p  -l read.txt -r read1.txt 192.168.40.143
```

The above command will upload the “*config*” file from the current directory on the target device to the directory previously configured on the TFTP server, under “*<WORKDIR>/tftpboot/*” and rename it to “*config_01*".

5.6. DHCP

DHCP (Dynamic Host Configuration Protocol) is a LAN network protocol. This refers to a range of IP addresses controlled by the server so that the client can automatically obtain the IP address and subnet mask assigned by the server when it logs in.

DHCP also includes two roles of server and client. In 4.1.1, we have tested the usage of DHCP client mode to automatically obtain IP address. When configuring WiFi AP mode in 4.1.2, we also tested the DHCP server mode to assign IP addresses to connected WiFi devices. Here is another way to get the IP address manually using the “dhclient” command and the “udhcpc” command, so that users can use it when debugging the network.

Connect the development board to the router with the CAT6 cable, manually assign the IP address to the eth0 network card using the DHCP client command, and observe the DHCP process to obtain the IP.

```
root@mys-8mmx:~# udhcpc -i eth0
udhcpc: started, v1.31.0
udhcpc: sending discover
udhcpc: sending select for 192.168.40.152
udhcpc: lease of 192.168.40.152 obtained, lease time 7200
/etc/udhcpc.d/50default: Adding DNS 223.5.5.5
/etc/udhcpc.d/50default: Adding DNS 201.104.111.114
```

Use the development board terminal as the dhcp client to dynamically obtain IP.

5.7. IPTables

Iptables is an administrative tool for IPv4 packet filtering and NAT. It is used to set up, maintain, and check the IP packet filtering rule table in the Linux 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 handle matching packets.

Devices using Linux systems typically use the “iptables” tool to configure firewalls. “iptables” handles packets based on methods defined by packet filtering rules, such as accept, reject, and drop. Let's use “iptables” to test intercepting ICMP packets, preventing other devices on the network from ping them. Specific commands to use see: https://linux.die.net/man/8/iptables.

1) Configure iptables for Target Device

Use the “iptables” configuration on the target device to discard the input ICMP package and not respond to ping probes from other hosts, with the following command:

```
root@mys-8mmx:~# iptables -A INPUT -p icmp --icmp-type 8 -j DROP
[16629.907561] audit: type=1325 audit(1608212303.192:10): table=filter family=
2 entries=0
[16629.915587] audit: type=1325 audit(1608212303.192:11): table=filter family=
2 entries=4
root@mys-8mmx:~# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j DROP
```

2) Ping the Target Device

Ping the target device on the development host and specifying a deadline of 10, the result is shown as follows:

```
PC$ ping 192.168.40.152 -w 10
```

```
PING 192.168.40.152 (192.168.40.152) 56(84) bytes of data.

--- 192.168.40.152 ping statistics ---
10 packets transmitted, 0 received, 100% packet loss, time 9210ms
```

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

3) Delete Rules for iptables

```
root@mys-8mmx:~# iptables -F
[16811.154441] audit: type=1325 audit(1608212484.436:12): table=filter family=
2 entries=5
root@mys-8mmx:~# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
```

4) Ping the Target Device Again

```
PC$ ping 192.168.40.152 -w 5
PING 192.168.40.152 (192.168.40.152) 56(84) bytes of data.
64 bytes from 192.168.40.152: icmp_seq=1 ttl=64 time=2.50 ms
64 bytes from 192.168.40.152: icmp_seq=2 ttl=64 time=2.57 ms
64 bytes from 192.168.40.152: icmp_seq=3 ttl=64 time=2.37 ms
64 bytes from 192.168.40.152: icmp_seq=4 ttl=64 time=2.35 ms
64 bytes from 192.168.40.152: icmp_seq=5 ttl=64 time=2.82 ms

--- 192.168.40.152 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 2.350/2.526/2.824/0.180 ms
```

Once the "iptables" rules are cleared, ping the target device again from the development host is ready to ping. The above example is just a simple demonstration, but iptables with various rules can be very powerful, and I won't go into the details here.

5.8. Ethtool

Ethtool is a tool for viewing and modifying Ethernet device parameters, which is useful during network debugging. Use this command to look at the Ethernet card information and try to modify its parameters.

We look at the help information for the command through “ethtool -h” .

```
root@mys-8mmx:~# ethtool --help
ethtool version 5.2
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 the basic information of the current device's Ethernet as follows.

```
root@mys-8mmx:~# ethtool eth0
Settings for eth0:
        Supported ports: [ TP MII ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
                                1000baseX/Full
        Supported pause frame use: Symmetric
        Supports auto-negotiation: Yes
        Supported FEC modes: Not reported
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
                                1000baseX/Full
        Advertised pause frame use: Symmetric
        Advertised auto-negotiation: Yes
        Advertised FEC modes: Not reported
        Link partner advertised link modes:  10baseT/Half 10baseT/Full
                                             100baseT/Half 100baseT/Full
                                             1000baseT/Full
        Link partner advertised pause frame use: Symmetric Receive-only
        Link partner advertised auto-negotiation: Yes
        Link partner advertised FEC modes: Not reported
        Speed: 1000Mb/s
        Duplex: Full
        Port: MII
        PHYAD: 4
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: d
        Link detected: yes
```

From the “ethtool” command, you can see that the current Ethernet supports 10, 100, and gigabit half-duplex and full-duplex modes. The current connection status is negotiated gigabit, full-duplex mode, MII interface, PHY address is 6, and so on.

We can also use the “ethtool” to set Ethernet parameters, which are useful for debugging and diagnosing Ethernet, such as forcing Ethernet to be set to 100 megabytes full duplex and turning off self-negotiation, as shown below：

```
root@mys-8mmx:~# ethtool -s eth0 speed 100 duplex full autoneg off
[17043.809978] fec 30be0000.ethernet eth0: Link is Down
[17043.815851] [dhd] CFG80211-ERROR) wl_cfg80211_netdev_notifier_call : wdev null. Do nothing
root@mys-8mmx:~# [17048.307837] fec 30be0000.ethernet eth0: Link is Up - 100Mbps/Full - flow control off
```

For more information about “ethtool”, please refer to the man page of “ethtool” at：http://man7.org/linux/man-pages/man8/ethtool.8.html .

5.9. iPerf3

IPerf3 is a tool that actively measures the maximum realized bandwidth over an IP network. It supports tuning various parameters such as test time, buffer size, and protocols (TCP, UDP, SCTP with IPv4 and IPV6). The iPerf3 can be divided into server mode or client mode by role, and we can use it to test and view network bandwidth in TCP mode, TCP window values, probability of retransmission, etc., as well as to test packet loss rates, latency, and jitter under specified UDP bandwidth.

We used an Ubuntu 16.04 system, a computer with gigabit card as the server of iPerf3, and the device under test as the client to test the performance on TCP and UDP for the device's Ethernet card.

Install iPerf3 on the server, as follows：

```
PC $ sudo apt-get install iperf3
```

Connect the server and device directly via CAT6 and configure their respective IP addresses. For example, we set the server IP to 192.168.40.143 and the device IP to 192.168.40.202, and use the ping command to test to make sure they are connected.

Note: Try not to connect to routers or switches to prevent test results from being affected by intermediate devices.

1) Test Performance under TCP Mode

- **Server Side（192.168.40.143)**

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

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

- **Client Side（192.168.40.202)**

Iperf3 works on the device as client, under TCP mode, where the parameters are described below：

- -c 192.168.40.143 : Work on the client side, connect to the server side 192.168.40.143
- -i 1:Test results are reported at an interval of 2 seconds
- -t 10 : The total test time is 10 seconds

```
root@mys-8mmx:~# iperf3  -c 192.168.40.143 -i 2 -t 10
Connecting to host 192.168.40.143, port 5201
[  5] local 192.168.40.100 port 49692 connected to 192.168.40.143 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.
```

After 10 seconds, the client finished the test and displayed the above test results, indicating that the TCP bandwidth was about 921Mbps, no retransmission, and the TCP window value was 554KBytes during the test.

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

```
PC$  iperf3 -s -i 2
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 192.168.40.100, port 49690
```

```
[  5] local 192.168.40.143 port 5201 connected to 192.168.40.100 port 49692
[ ID] Interval           Transfer     Bandwidth
[  5]   0.00-2.00   sec   215 MBytes   901 Mbits/sec
[  5]   2.00-4.00   sec   219 MBytes   918 Mbits/sec
[  5]   4.00-6.00   sec   221 MBytes   925 Mbits/sec
[  5]   6.00-8.00   sec   221 MBytes   927 Mbits/sec
[  5]   8.00-10.00  sec   219 MBytes   920 Mbits/sec
[  5]  10.00-10.00  sec   283 KBytes   920 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       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
-----------------------------------------------------------
Server listening on 5201
```

2) Test Performance under UDP Mode

- **Server Side（192.168.40.143)**

Continue running iPerf3 on the server and use the -s parameter to indicate that you are working in the server mode.

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

- **Client Side（192.168.40.202)**

Iperf3 works on the device as client, under UDP mode, where the parameters are described below:

- -u : Work in UDP mode
- -c 192.168.40.143 : Work on the client side, connect to the server side 192.168.40.143
- -i 1 : Test results are reported at an interval of 1 seconds
- -t 10 :The total test time is 10 seconds
- -b 100M : Set the UDP transmission bandwidth as 100Mbps

```
root@mys-8mmx:~# iperf3 -u -c 192.168.40.143 -i 1 -t 10 -b 100M
```

```
Connecting to host 192.168.40.143, port 5201
[  5] local 192.168.40.202 port 32972 connected to 192.168.40.143 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-1.00   sec  11.9 MBytes  99.9 Mbits/sec  8626
[  5]   1.00-2.00   sec  11.9 MBytes   100 Mbits/sec  8633
[  5]   2.00-3.00   sec  11.9 MBytes   100 Mbits/sec  8632
[  5]   3.00-4.00   sec  11.9 MBytes   100 Mbits/sec  8633
[  5]   4.00-5.00   sec  11.9 MBytes   100 Mbits/sec  8632
[  5]   5.00-6.00   sec  11.9 MBytes   100 Mbits/sec  8633
[  5]   6.00-7.00   sec  11.9 MBytes   100 Mbits/sec  8632
[  5]   7.00-8.00   sec  11.9 MBytes   100 Mbits/sec  8633
[  5]   8.00-9.00   sec  11.9 MBytes   100 Mbits/sec  8633
[  5]   9.00-10.00  sec  11.9 MBytes   100 Mbits/sec  8632
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.000 ms  0/86319
(0%)  sender
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.166 ms  0/86317
(0%)  receiver

iperf Done.
```

The client completes the test after 10 seconds and displays the above test results, showing that UDP did not lose packets when specifying a bandwidth of 100Mbps.

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

```
$ $ iperf3 -s -i 2
-----------------------------------------------------------
Accepted connection from 192.168.40.202, port 47348
[  5] local 192.168.40.143 port 5201 connected to 192.168.40.202 port 32972
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
```

```
[  5]   0.00-2.00   sec  23.8 MBytes  99.8 Mbits/sec  0.230 ms  0/17236
(0%)
[  5]   2.00-4.00   sec  23.8 MBytes   100 Mbits/sec  0.206 ms  0/17263
(0%)
[  5]   4.00-6.00   sec  23.8 MBytes   100 Mbits/sec  0.210 ms  0/17267
(0%)
[  5]   6.00-8.00   sec  23.8 MBytes   100 Mbits/sec  0.229 ms  0/17265
(0%)
[  5]   8.00-10.00  sec  23.8 MBytes   100 Mbits/sec  0.225 ms  0/17267
(0%)
[  5]  10.00-10.00  sec  26.9 KBytes   109 Mbits/sec  0.166 ms  0/19 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datag
rams
[  5]   0.00-10.00  sec   119 MBytes   100 Mbits/sec  0.166 ms  0/86317
(0%)
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

The client adjust the “-b” parameter and continues to increase the specified UDP bandwidth. When packet loss occurs, the UDP bandwidth measured is the actual UDP bandwidth.If the specified bandwidth reaches the Ethernet card's highest bandwidth of 1Gbps and still no packets are lost, the bandwidth returned by the iPerf3 test is the actual UDP bandwidth. For example, the following example specifies a bandwidth of 1000Mbps, still without packet loss, but the actual bandwidth is around 600Mbps.

```
root@mys-8mmx:~# iperf3 -u -c 192.168.40.143 -i 1 -t 10 -b 1000M
Connecting to host 192.168.40.143, port 5201
[  5] local 192.168.40.202 port 47957 connected to 192.168.40.143 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-1.00   sec   118 MBytes   993 Mbits/sec  85764
[  5]   1.00-2.00   sec   118 MBytes   992 Mbits/sec  85638
[  5]   2.00-3.00   sec   118 MBytes   990 Mbits/sec  85500
[  5]   3.00-4.00   sec   118 MBytes   991 Mbits/sec  85524
[  5]   4.00-5.00   sec   118 MBytes   993 Mbits/sec  85727
[  5]   5.00-6.00   sec   120 MBytes  1.01 Gbits/sec  86934
[  5]   6.00-7.00   sec   120 MBytes  1.01 Gbits/sec  87108
[  5]   7.00-8.00   sec   120 MBytes  1.01 Gbits/sec  87196
[  5]   8.00-9.00   sec   120 MBytes  1.01 Gbits/sec  87208
[  5]   9.00-10.00  sec   119 MBytes   995 Mbits/sec  85906
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagra
ms
[  5]   0.00-10.00  sec  1.16 GBytes   999 Mbits/sec  0.000 ms  0/862505
(0%)  sender
[  5]   0.00-10.00  sec  1.07 GBytes   920 Mbits/sec  0.018 ms  67869/86217
6 (7.9%)  receiver

iperf Done.
```

Iperf3 also has a number of parameters that can be configured during the test so that users can tailor the test to their actual application needs. For example, you can increase the value of the “-t” parameter for long time stress test, or specify the “-P” parameter for multiple concurrent stress tests. For more information about the IPerf3 test, please refer to https://iperf.fr/iperdoc. Ph #3doc.

6. Linux Graphics System

6.1. GPU

MYS-8MMX have 2 GPU cores, one for 3 D data processing and the other for 2D and 3 D acceleration. 3D GPU Support:

- OpenGL ES 1.1 ， 2.0
- Open VG 1.1
- 2D GPU Support
- Mixed planes

1) OpenGL ES2.0

The factory default image (myir-image-full) already has packaged test examples, go to the */opt/imx-gpu-sdk/GLES2* directory, you can find them,for example：

```
root@mys-8mmx:~# cd /opt/imx-gpu-sdk/GLES2
root@mys-8mmx:/opt/imx-gpu-sdk/GLES2# ls
Bloom                DFSimpleUI101     ModelLoaderBasics     S04_Projection
                 Stats
Blur                EightLayerBlend   ModelViewer           S06_Texturing
              T3DStressTest
DFGraphicsBasic2D   FractalShader     S01_SimpleTriangle    S07_EnvironmentMa
pping               TextureCompression
DFNativeBatch2D     InputEvents       S02_ColoredTriangle   S08_EnvironmentM
appingRefraction    VIVDirectTextureMultiSampling
DFSimpleUI100       LineBuilder101    S03_Transform         S09_VIV_direct_text
ure
```

You can use the S08_EnvironmentMappingRefraction example for testing：

```
root@mys-8mmx:/opt/imx-gpu-sdk/GLES2# ./S08_EnvironmentMappingRefractio
n/GLES2.S08_EnvironmentMappingRefraction_Wayland
```

In addition, there are glmark2-es2-wayland program that glmark comes with, and QT examples under */user/share/example/OpenGL* can be tested.

2) OpenVG

The factory default image (myir-image-full) already has packaged test examples, go to the */opt/imx-gpu-sdk/ OpenVG* directory, you can find them,for example：

```
root@mys-8mmx:/opt/imx-gpu-sdk/OpenVG# ls
BitmapFont   CoverFlow   DFGraphicsBasic2D   Example1   Example2   Example3
 SimpleBench   VGStressTest
```

You can use the OpenVG.Example3_Wayland example for testing：

```
root@mys-8mmx:/opt/imx-gpu-sdk/OpenVG# ./Example3/OpenVG.Example3_Wayland
```

6.2. Wayland + Weston + QT

The MYS-8MMX devices only support wayland and Xwayland, not support fb mode, wayland use weston graphics desktop, you can run the QT program on the weston desktop.

1) View Process Services

```
root@mys-8mmx:~# pstree
systemd-+-agetty
        |-atd
        |-avahi-daemon---avahi-daemon
        |-crond
        |-dbus-daemon
        |-haveged
        |-klogd
        |-login---sh---pstree
        |-ntpd---{ntpd}
        |-ofonod
        |-proftpd
        |-rpc.statd
        |-rpcbind
        |-syslogd
        |-systemd---(sd-pam)
        |-systemd-journal
        |-systemd-logind
        |-systemd-network
        |-systemd-resolve
        |-systemd-timesyn---{systemd-timesyn}
        |-systemd-udevd
        |-tee-supplicant
        `-weston-+-(sd-pam)
                 |-weston-desktop-
```

```
                    |-weston-keyboard
                    `-{weston}
```

- ➢ weston-desktop   weston Desktop
- ➢ weston-keyboard   weston keyboard support

2) View System Environment

```
root@mys-8mmx:~#  env
SHELL=/bin/sh
EDITOR=vi
PWD=/home/root
LOGNAME=root
XDG_SESSION_TYPE=tty
HOME=/home/root
LANG=C
QT_QPA_PLATFORM=wayland
XDG_SESSION_CLASS=user
TERM=xterm
USER=root
SHLVL=1
XDG_SESSION_ID=c2
XDG_RUNTIME_DIR=/run/user/0
PS1=\u@\h:\w\$
TSLIB_TSDEVICE=/dev/input/touchscreen0
HUSHLOGIN=FALSE
PATH=/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin
DBUS_SESSION_BUS_ADDRESS=unix:path=/run/user/0/bus
MAIL=/var/spool/mail/root
_=/usr/bin/env
OLDPWD=/opt/imx-gpu-sdk/OpenVG
```

- ➢ XDG_RUNTIME_DIR： Running environment
- ➢ QT_QPA_PLATFORM： Running platform

3) Run QT routine

QT examples are in the /user/share/example directory, running routine:

```
root@mys-8mmx:~# /usr/share/examples/widgets/touch/pinchzoom/pinchzoom
```

You can see the QT interface on the screen. If you want to run the QT program at boot time, you need to add two variables to the script:XDG_RUNTIME_DIR and QT_QPA_PLATFORM，as follows:

```
root@mys-8mmx:~# cat /etc/start_service.sh
#!/bin/sh
echo "start service" > /var/startservice.log

export XDG_RUNTIME_DIR=/run/user/0
export QT_QPA_PLATFORM=wayland
sleep 2
/usr/share/examples/widgets/touch/pinchzoom/pinchzoom
sleep 10
/home/root/power-check.sh &
exit 0
```

7. Multimedia Applications

7.1. Camera

This section uses the system's own gstreamer, V4L2-utils, media-ctl commands to evaluate and test the CSI camera(Type:MY-CAM003M), which is listed in Table 1-1. We mainly preview, capture frame (take photos) with the CSI camera.

1) Check Ov5640 Camera Information

- **View the Device Node**

When the CSI camera is connected through j11, the character device node /dev/video0 will be generated：

```
root@mys-8mmx:~# ls /dev/video0
/dev/video0
```

- **Use v4l2-ctl to View the Supported Resolution of /dev/video0**

```
root@mys-8mmx:~# v4l2-ctl -D -d /dev/video0  --list-formats-ext
Driver Info:
        Driver name        : mx6s-csi
        Card type          : i.MX6S_CSI
        Bus info           : platform:32e20000.csi1_bridge
        Driver version     : 5.4.3
        Capabilities       : 0x84200001
                Video Capture
                Streaming
                Extended Pix Format
                Device Capabilities
        Device Caps        : 0x04200001
                Video Capture
                Streaming
                Extended Pix Format
Frame rate set to 30.000 fps
```

```
ioctl: VIDIOC_ENUM_FMT
        Type: Video Capture

        [0]: 'YUYV' (YUYV 4:2:2)
                Size: Discrete 640x480
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 320x240
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x480
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1280x720
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1920x1080
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 2592x1944
                        Interval: Discrete 0.067s (15.000 fps)
```

From the above log, we can clearly see the supported frame rate under the corresponding resolution, such as the supported frame rate of 320x240 is 30fps.

2) Camera Preview

- **Use gstreamer for Preview**

Using the “gst-launch” command in gstreamer for a preview of the camera, the terminal executes the following command：

```
root@mys-8mmx:~# gst-launch-1.0 v4l2src ! "video/x-raw, width=640, Height=
480, framerate=(fraction)30/1" ! queue !  glimagesink
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Got context from element 'sink': gst.gl.GLDisplay=context, gst.gl.GLDisplay=(Gs
tGLDisplay)"\(GstGLDisplayWayland\)\ gldisplaywayland0";
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[  114.775042] ov5640_mipi 1-003c: s_stream: 1
```

Once executed, the screen generates a preview window with a width of 640, a height of 480, and a frame rate of 30 frames per second.

- **Use gstreamer for Capture**

Use the “gst-launch” command in gstreamer to take a photo of the camera and save it as a JPEG image. The terminal executes the following command：

```
root@mys-8mmx:~# gst-launch-1.0  v4l2src device=/dev/video0 num-buffers=
1 ! jpegenc ! filesink location=/home/root/test.jpg
Setting pipeline to PAUSED ...
Pipeline is live and does not need PREROLL ...
Setting pipeline to PLAYING ...
New clock: GstSystemClock
[  195.804784] ov5640_mipi 1-003c: s_stream: 1
[  196.159492] skip frame 1
Got EOS from element "pipeline0".
Execution ended after 0:00:03[  198.809129] ov5640_mipi 1-003c: s_stream: 0
.413483625
Setting pipeline to PAUSED ...
Setting pipeline to READY ...
Setting pipeline to NULL ...
Freeing pipeline ...
root@mys-8mmx:~# weston-image test.jpg
could not load cursor 'dnd-move'
could not load cursor 'dnd-copy'
could not load cursor 'dnd-none'
```

The above command means save frame 1 and convert to JPEG.Because the current camera only supports pixels in "YUYV" format, it needs to be transcoded into JPEG format. The parameters are described as follows:

- Jpegenc：Converted image in JPEG format

7.2. VPU

VPU(video processing unit) is a new core engine of video processing platform, which has the ability of hard decoding and reducing CPU load

The MYS-8MMX device supports VPU video encoding and decoding. VPU support the following decoding formats:

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

The supporting coding format is as follows:

- H.264
- VP8

1) Decoding Test

Next, use the GST command to obtain the audio and video information of the file, and play it to the screen to test the decoding function of VPU.

- **Video Source**

The following steps is an example on how to play video on a U disk.

```
root@mys-8mmx:~/video# [  296.383528] usb 1-1.1: USB disconnect, device number 3
[  296.522321] FAT-fs (sda1): unable to read boot sector to mark fs as dirty
[  298.674737] usb 1-1.1: new high-speed USB device number 4 using ci_hdrc
[  298.811447] usb-storage 1-1.1:1.0: USB Mass Storage device detected
[  298.818103] scsi host0: usb-storage 1-1.1:1.0
[  299.840510] scsi 0:0:0:0: Direct-Access     Kingston DataTraveler 3.0     PQ: 0 ANSI: 6
[  299.853245] sd 0:0:0:0: [sda] 60437492 512-byte logical blocks: (30.9 GB/28.8 GiB)
[  299.861706] sd 0:0:0:0: [sda] Write Protect is off
[  299.869494] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or FUA
[  299.911606]  sda: sda1
[  299.918209] sd 0:0:0:0: [sda] Attached SCSI removable disk

root@mys-8mmx:~/video# cd /run/media/sda1/video/
root@mys-8mmx:/run/media/sda1/video# ls -l
total 79056
-rwxrwx--- 1 root disk  6760712 Dec 10 14:29 h264_bframe_crash.mp4
-rwxrwx--- 1 root disk 33184878 Dec 10 14:46 halo2_wmp9_WMV3+audio0x162.wmv
-rwxrwx--- 1 root disk 35698163 Dec 10 14:47 vandread_ep13_op_wm8_ver.wmv
-rwxrwx--- 1 root disk  5291494 Dec 10 14:10 video-h265.mkv
```

- **View Video Format**

Use the gst-discoverer-1.0 command to obtain audio and video formats information about the video-h265.mkv video example.

```
root@mys-8mmx:/run/media/sda1/video# gst-discoverer-1.0 video-h265.mkv
......................................
Topology:
  container: Matroska
    audio: MPEG-4 AAC
    video: H.265 (Main Profile)

Properties:
  Duration: 0:00:16.160000000
  Seekable: yes
  Live: no
  Tags:
      codec: AAC, H.265/HEVC
```

```
        language code: eng
        container format: Matroska
        application name: HandBrake 0.10.0 2014112200
        video codec: H.265/HEVC
        audio codec: AAC
```

- audio: MPEG-4 AAC   Audio format AAC
- video: H.265   Video format H.265
- container: Matroska   Matroska with filter containers

- **Play Video**

There is a general command gst-play-1.0 for playing video, which can play audio and video, but the operation space is small. The other is gst-launcher-1.0, which has flexible functions.

Play video using gst-play-1.0:

```
root@mys-8mmx:/run/media/sda1/video# gst-play-1.0 video-h265.mkv
```

Play video using gst-luancher-1.0:

```
root@mys-8mmx:/run/media/sda1/video# gst-launch-1.0 filesrc location=video-h265.mkv typefind=true ! video/x-matroska ! aiurdemux ! queue ! vpudec ! autovideosink
```

- filesrc location：Specifies the source and source file paths to play
- typefind： Represents a known file type
- video/x-matroska： Matroska containers are used
- aiurdmux： Audio and video analyzer demux
- queue：   Using queues
- vpudec：   Decoder VPUDEC
- Autovideosink：Select output source by default, optional waylandsink and glimagesink

2) Coding Test

The MYS-8MMX device supports vpuenc_ H264 and vpuenc_ VP8 encoder.In addition, MVC extension to H.264 is also supported.The following steps are an example of how to encode H.265 format video into VP8 format video.

- **Encoding h.265 video into VP8 video**

According to the above example, it is known that video-h265.mkv is in h.265 format. Execute the following command to encode the video:

```
root@mys-8mmx:/run/media/sda1/video# gst-launch-1.0 -e filesrc location=video-h265.mkv typefind=true ! video/x-matroska ! aiurdemux ! vpudec  ! vpuenc_vp8 ! matroskamux ! filesink location=test.mkv
```

- -e：Add a terminator when an exception is interrupted
- vpuenc_vp8: VP8 hardware encoder
- matroskamux: Audio-Video Mixer
- filesink:Output by file，location：Represents the save path

```
root@mys-8mmx:/run/media/sda1# gst-discoverer-1.0 test.mkv
Analyzing file:///run/media/sda1/test.mkv

====== AIUR: 4.5.4 build on Sep  5 2020 04:45:35. ======
        Core: MKVPARSER_01.08.02  build on Feb 15 2019 09:59:27
 file: /usr/lib/imx-mm/parser/lib_mkv_parser_arm_elinux.so.3.1
------------------------
    Track 00 [video_0] Enabled
        Duration: 0:01:01.966000000
        Language: eng
    Mime:
        video/x-vp8, width=(int)544, height=(int)960, framerate=(fraction)30/1
------------------------
====== VPUDEC: 4.5.4 build on Sep  5 2020 04:45:48. ======
        wrapper: 3.0.0 (VPUWRAPPER_ARM64_LINUX Build on Sep  5 2020 02:26:35)
        vpulib: 1.1.1
        firmware: 1.1.1.0
Done discovering file:///run/media/sda1/test.mkv

Topology:
  container: Matroska
    video: VP8

Properties:
  Duration: 0:01:01.966000000
  Seekable: yes
  Live: no
  Tags:
      codec: VP8
      language code: eng
      container format: Matroska
      application name: GStreamer Matroska muxer
      video codec: VP8
```

Then use the gst-play-1.0 tool to play：

```
root@mys-8mmx:~# gst-play-1.0 test.mkv
```

8. System Tools

Some commonly used system tools are included in the default image of the factory, which is convenient for users to view and manage various resources of the system in system debugging or actual deployed products, and can also be called in SHELL scripts or other applications. These tools may not fully meet the needs of the user's system customization, at which point the system developer needs to make appropriate adjustments based on the actual situation.

8.1. Compress and Decompress Tools

This section mainly tests the compression and decompression tools of the system. Compression is to compress multiple files into a compression package , then it will be convenient for file transfer. And decompression will make the compressed files back to the original size for easy use. This section illustrates the file system with tools such as “tar”, “gzip”, “gunzip”, and so on.

1) tar Tool

- **Syntax Format**

The “tar” tool, now commonly used in Linux, not only packages files, but also compresses, views, adds, and unzips them. Here is the package operation.

```
root@mys-8mmx:~#  tar --help
Usage: tar [OPTION...] [FILE]...
GNU 'tar' saves many files together into a single tape or disk archive, and ca
n
restore individual files from the archive.

Examples:
  tar -cf archive.tar foo bar  # Create archive.tar from files foo and bar.
  tar -tvf archive.tar         # List all files in archive.tar verbosely.
  tar -xf archive.tar          # Extract all files from archive.tar.
```

The detailed parameters are described below:

- -c : Create a compressed file parameter instruction
- -x : Unpacking a compressed file parameter instruction
- -t : Check out the files in a tar file! Note in particular that c/x/t can only exist in one parameter. Do not exist at the same time! Because it is impossible to compress and uncompress at the same time.
- -z : Do you have gzip properties at the same time? Is gzip compression required?
- -j : Do you have a bzip2 attribute at the same time? Is bzip2 compression necessary?
- -v : Show files during compression! This is often used, but is not recommended when running in the background
- -f : Use file name, please note, the file name must be followed -f parameter immediately
- -p : Use the original properties of the original file (properties do not change depending on the user)
- -P : Can use absolute path to compress
- --exclude FILE: During compression, do not pack a FILE

- **Use tar to Compress**

Create a new test.txt file and type the following command to package the file in “.gz” format:

```
root@mys-8mmx:~#  tar -czf test.tar.gz test.txt
root@mys-8mmx:~# ls -l read.tar.gz
-rw-r--r-- 1 root root 133 Dec 17 17:13 read.tar.gz
```

With the z parameter, the “.tar.gz” or “.tgz” represents a gzip-compressed tar file.

- **Use tar to Decompress**

Unpack the file which has been compressed as tar.gz.

```
root@mys-8mmx:~#  tar -xvf test.tar.gz
test.txt
root@mys-8mmx:~#  ls
OpenAMP_TTY_echo.elf  rs485_write  test.tar.gz  uart_test
rs485_read            test.c       test.txt     wifi.conf
```

2) gzip Tool

- **Syntax Format**

"gzip" is a file compression and decompression command frequently used in Linux systems, both convenient and easy to use.

```
root@mys-8mmx:~#  gzip --help
Usage: gzip [OPTION]... [FILE]...
```

- **Use gzip to Compress**

```
root@mys-8mmx:~#  gzip test.txt
root@mys-8mmx:~#  ls
OpenAMP_TTY_echo.elf  rs485_write  test.tar.gz  uart_test
rs485_read            test.c       test.txt.gz  wifi.conf
```

- **Use gunzip to Decompress**

```
root@mys-8mmx:~#  gunzip test.txt.gz
root@mys-8mmx:~#  ls
OpenAMP_TTY_echo.elf  rs485_write  test.tar.gz  uart_test
rs485_read            test.c       test.txt     wifi.conf
```

8.2. File System Tools

The main test system file system tools, this section will introduce several common file system management tools. System comes with file system tools "mount", "mkfs", "fsck", "dumpe2fs".

1) mount Tool

"mount" is a command under Linux that connects a partition to a folder on Linux, associating the partition with that directory, so simply accessing the folder is equivalent to accessing the partition, in the following syntax format:

```
root@mys-8mmx:~# 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.
```

- **Mounted U disk**

```
root@mys-8mmx:~# mount /dev/sda1 /mnt/
```

2) mkfs Tool

After partitioning the hard disk, the next step is to set up the Linux file system. Similar to a formatted hard disk in Windows. Setting up a file system on a hard disk partition washes out the data on the partition and is not recoverable, so make sure that the data on the partition is no longer used before setting up the file system. The command to set up the file system is mkfs, in the following syntax format:

```
root@mys-8mmx:~# 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).
```

- **Format U disk**

```
root@mys-8mmx:~#  umount /mnt
root@mys-8mmx:~#  umount /run/media/sda1/

root@mys-8mmx:~#  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 old-
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 Tool

The "fsck" command is mainly used to check the correctness of the file system. When the file system fails, the fsck instruction can be used to try to fix it. And fix the Linux disk. For example：

```
root@mys-8mmx:~#  fsck -a /dev/sda1
fsck from util-linux 2.34
/dev/sda1: clean, 11/972944 files, 86964/3890688 blocks
```

4) dumpe2fs Tool

Prints information about super blocks and blocks groups of existing file systems on a particular device. The development board enters the following command to see the application syntax：

```
root@mys-8mmx:~#  dumpe2fs -h
dumpe2fs 1.45.3 (14-Jul-2019)
Usage: dumpe2fs [-bfghimxV] [-o superblock=<num>] [-o blocksize=<num>]
device
```

View detailed properties of the file system formatted, for example, by entering a command to view the details of a disk：

```
root@mys-8mmx:~#  dumpe2fs -h
dumpe2fs 1.45.3 (14-Jul-2019)
Usage: dumpe2fs [-bfghimxV] [-o superblock=<num>] [-o blocksize=<num>]
device
```

```
root@mys-8mmx:~#
root@mys-8mmx:~#  root@mys-8mmx:~#  dumpe2fs -h^C
root@mys-8mmx:~#  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, so when the disk is formatted, the operating system automatically divides the disk into two areas. One is the data area, where file data is stored; The other is the inode section (inode table), which holds the information contained in the inode.

```
root@mys-8mmx:~#  dumpe2fs /dev/sda1 | grep -i "inode size"
dumpe2fs 1.45.3 (14-Jul-2019)
Inode size:               256
```

Look at the number of blocks on a disk. When the operating system reads the hard disk, it will not read sectors one by one, which is too inefficient. Instead, it will read multiple sectors in a row at one time, that is, read a "block" at one time. This "block" of multiple sectors is the smallest unit of file access.

```
root@mys-8mmx:~#  dumpe2fs /dev/sda1 | grep -i "block size"
dumpe2fs 1.45.3 (14-Jul-2019)
Block size:               4096
```

8.3. Disk Management Utils

This section mainly test system disk management tools, including several common disk management tools. The system comes with disk management tools “fdisk” , “dd” , “du” , “df” , and so on, These commands allow you to monitor your daily disk usage.

1) fdisk Disk Partitioning Tool

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

```
root@mys-8mmx:~# fdisk -h
Usage:
 fdisk [options] <disk>       change partition table
 fdisk [options] -l [<disk>] list partition table(s)
```

Partition the eMMC as follows：

```
root@mys-8mmx:~# fdisk /dev/mmcblk2p2

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

The old ext4 signature will be removed by a write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0x43403b02.

Command (m for help):
```

2) dd tool

The “dd” command is used to copy the specified input file to the specified output file. And the format can be converted during replication. The difference between the “dd” command and the “cp” command is that the “dd” command can be done on a floppy disk that has not been created, and the data copied to the floppy disk is actually a mirror 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 nam
e>] [bs=block size(byte)] [count=block count]
```

An example of creating a file of 2M size is shown below:

```
root@mys-8mmx:~#  time dd  if=/dev/zero of=ffmpeg1 bs=2M count=1 con
v=fsync
1+0 records in
1+0 records out
2097152 bytes (2.1 MB, 2.0 MiB) copied, 0.116105 s, 18.1 MB/s

real      0m0.126s
user      0m0.001s
sys       0m0.019s
```

3) du Tool

The du command is used to display disk space usage. This command shows, level by level, how each level of subdirectory of the specified directory occupies a block of file system data. “du” is generally used in the following syntax:

```
root@mys-8mmx:~#  du --help
Usage: du [OPTION]... [FILE]...
  or:  du [OPTION]... --files0-from=F
Summarize disk usage of the set of FILEs, recursively for directories.
```

Partial parameter are as follows:

- -a: Displays the size of all directories or files

- -h: Take K,M and G Bytes as units to improve the readability of inform ation
- -k: Output in Kilo Bytes
- -m: Output in Mega Bytes

Statistics the file size generated by the dd command：

```
root@mys-8mmx:~#  du ffmpeg1
2048    ffmpeg1
root@mys-8mmx:~#  du -h ffmpeg1
2.0M    ffmpeg1
```

4) df Tool

Used to display disk usage statistics for the current file system on a Linux system, commonly used as follows：

```
root@mys-8mmx:~#  df --help
Usage: df [OPTION]... [FILE]...
Show information about the file system on which each FILE resides,
or all file systems by default.
```

Partial parameter are as follows：

- -h： The appropriate units can be displayed based on the size used
- -i: View the number of inodes and inode usage under the partition
- -T: Print out the file system type

To see the number of inodes and inode usage under the partition, use the following command：

```
root@mys-8mmx:~# df -i
Filesystem      Inodes IUsed  IFree IUse% Mounted on
/dev/root       471888 37956 433932    9% /
devtmpfs        166136   465 165671    1% /dev
tmpfs           248784     1 248783    1% /dev/shm
tmpfs           248784   668 248116    1% /run
tmpfs           248784    13 248771    1% /sys/fs/cgroup
tmpfs              248784      15  248769      1%  /tmp
tmpfs              248784      30  248754      1%  /var/volatile
/dev/mmcblk2p1          0       0       0       -  /run/media/mmcblk2p1
tmpfs              248784       9  248775      1%  /run/user/0
```

The inode is defined by the system during formatting, and the inode depends on the size of the disk partition. When we reach 100 percent inode usage, we cannot write data to disk even though we still have disk space left. Please refer to Section 2.5 for other application examples.

8.4. Process Management Utils

Process is also an important concept in the operating system, it is a process of the execution of a program, the program is a static description of the process, the system run every program is in its process run. All processes in Linux are interconnected, and all processes have a parent except for initializing the process. Instead of being created, new processes are copied or copied from previous processes. All processes in Linux are derived from an “init” process with process number 1. Linux consists of three different types of processes, each with its own characteristics and attributes:

- The Interaction Process: A process started by a Shell can run both in the foreground and in the background.
- The Batch Process: This process has no connection to the terminal and is a sequence of processes. This process is committed to a process waiting for execution in queue order.
- Monitor Processes (daemons): Daemons are always active and usually run in the background, and daemons are usually started automatically by the system at the beginning through script activation or root.

For Linux system, process management is an important link, and process management is usually achieved through process management tools. There are several commonly used process management commands in Linux system: “ps”, “top”, “vmstat”, “kill” and so on.

1) ps Tool

- **Syntax Format**

“ps” is a command to show the status of current processes. The general syntax is as follows:

```
root@mys-8mmx:~#  ps --help

Usage:
 ps [options]
 Try 'ps --help <simple|list|output|threads|misc|all>'
  or 'ps --help <s|l|o|t|m|a>'
 for additional help text.

For more details see ps(1).
```

Partial parameter are as follows:

- -u: Organize a user-centric display of process status information.
- -a: A process that is not terminal dependent.
- -x: The process associated with the terminal.

Usually the above commands are combined with: aux

- -e: Show all processes; The equivalent of ax.
- -f: Displays full format program information.

Usually the above commands are combined with: ef

- -H: Shows the number of processes at the process level
- -F: Display more program information

Usually the above commands are combined with: eHF

- **View information on all processes**

An example of displaying information for all processes is shown below:

```
root@mys-8mmx:~#  ps -aux
USER           PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME
COMMAND
root             1  0.0  0.3  92272  7176 ?        Ss   Sep04   0:03 /sbin/init
root             2  0.0  0.0      0     0 ?        S    Sep04   0:00 [kthreadd]
root             3  0.0  0.0      0     0 ?        I<   Sep04   0:00 [rcu_gp]
root             4  0.0  0.0      0     0 ?        I<   Sep04   0:00 [rcu_par_g
p]
```

```
root          8  0.0  0.0       0       0 ?        I<   Sep04    0:00  [mm_perc
pu_wq]
root          9  0.0  0.0       0       0 ?        S    Sep04    0:00  [ksoftirqd/
0]
root         10  0.0  0.0       0       0 ?        I    Sep04    0:00  [rcu_preem
pt]
root         11  0.0  0.0       0       0 ?        S    Sep04    0:00  [migration
/0]
root         12  0.0  0.0       0       0 ?        S    Sep04    0:00  [cpuhp/0]
root         13  0.0  0.0       0       0 ?        S    Sep04    0:00  [cpuhp/1]
root         14  0.0  0.0       0       0 ?        S    Sep04    0:00  [migration
/1]
```

A brief explanation of some of the taskbars above：

- VSZ：Virtual memory size
- RSS：Resident size
- STAT：There are several process states
  - R：Running
  - S：Interruptable sleeping
  - D：Uninterruptable sleeping
  - T：Stopped
  - Z: Zombie process
  - +: Foreground process
  - N: Low priority process
  - l: Multithreaded process

2) top Tool

- **Syntax Format**

The “top” command puts quite a bit of overall system performance information on one screen. The display can also be changed in an interactive manner. Dynamic continuous monitoring of the running state of the process, the "top" syntax is generally as follows:

```
root@mys-8mmx:~#   top -help
  procps-ng 3.3.15
Usage:
  top -hv | -bcEHiOSs1 -d secs -n max -u|U user -p pid(s) -o field -w [cols]
```

See section 2.1 for an example of dynamically viewing system processes.

- **View System Dynamic Processes**

```
root@mys-8mmx:~#  top
top - 02:03:28 up  2:28,  1 user,  load average: 0.01, 0.35, 0.99
Tasks: 135 total,   1 running, 134 sleeping,   0 stopped,   0 zombie
%Cpu(s):  1.4 us,  1.4 sy,  0.0 ni, 97.1 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1943.7 total,   1487.8 free,    244.2 used,    211.6 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   1597.4 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME
+ COMMAND

    982 root      20   0    3444   2024   1616 R   5.9   0.1   0:00.02 top

      1 root      20   0   92272   7176   5064 S   0.0   0.4   0:03.49 syste
md

      2 root      20   0       0      0      0 S   0.0   0.0   0:00.03 kthrea
dd

      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
```

3) vmstat Tool

- **Syntax Format**

This command looks at the usage status of the memory space and gives you a snapshot of the performance of the entire system. “vmstat” runs in two modes: sample mode and average mode. If no parameters are specified, “vmstat” statistics run in average mode and “vmstat” displays the average of all statistics since system startup. The common syntax and parameters are as follows:

```
root@mys-8mmx:~#  vmstat  -h
Usage:
 vmstat [options] [delay [count]]

Options:
 -a, --active               active/inactive memory
 -f, --forks                number of forks since boot
 -m, --slabs                 slabinfo
 -n, --one-header            do not redisplay header
 -s, --stats                event counter statistics
 -d, --disk                 disk statistics
 -D, --disk-sum              summarize disk statistics
 -p, --partition <dev>      partition specific statistics
 -S, --unit <char>          define display unit
 -w, --wide                 wide output
 -t, --timestamp            show timestamp

 -h, --help     display this help and exit
 -V, --version  output version information and exit

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

- **vmstat runs in average mode**

“vmstat” runs in average mode, showing the average of all statistics since system startup:

```
root@mys-8mmx:~# vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 st
 0  0      0 1616592   7296 113264    0    0    52     1  403  468 13  4
83  0  0
```

A brief explanation of some of the taskbars above:

- r: Number of processes currently running. Instead of waiting for I/O, these processes have ready numbers to run. Ideally, the number of processes that can run is equal to the number of CPU available
- b: The number of blocked processes waiting for I/O to complete
- forks: The number of times a new process was created
- in: The number of system interrupts
- cs: The number of times a context switch occurred in the system
- us: Percentage of total CPU time consumed by user processes
- sy: The percentage of total CPU time consumed by the system code, which includes time consumed in the System, IRQ, and Softirq states
- wa: The percentage of total CPU consumed waiting for I/O
- id: The percentage of total CPU time consumed by the system idle

- **Statistical system data details**

Statistical system various data details are as follows:

```
root@mys-8mmx:~# vmstat  -s
      1990276 K total memory
       252856 K used memory
        73608 K active memory
        77500 K inactive memory
      1616720 K free memory
         7296 K buffer memory
       113404 K swap cache
```

```
            0 K total swap
            0 K used swap
            0 K free swap
        23675 non-nice user cpu ticks
            0 nice user cpu ticks
         6236 system cpu ticks
       157486 idle cpu ticks
          130 IO-wait cpu ticks
          976 IRQ cpu ticks
          386 softirq cpu ticks
            0 stolen cpu ticks
        92292 pages paged in
         2552 pages paged out
            0 pages swapped in
            0 pages swapped out
       760095 interrupts
       881458 CPU context switches
   1608225090 boot time
          696 forks
```

A brief explanation of some of the information above：

- total memory：Total system memory
- used memory：Used memory
- CPU ticks：This shows the CPU time since the system started, where "tick" is a unit of time
- forks：Roughly speaking, this represents the number of new processes that have been created since system startup

“vmstat” provides a great deal of information about the performance of Linux systems. It is one of the core tools for investigating system problems.

4) kill tool

- **Syntax Format**

Sends the specified signal to the appropriate process. Not specifying the model sends SIGTERM (15) to terminate the specified process. If the program cannot be terminated with the available "-KILL" parameter, the signal it sends is SIGKILL(9), which forces the process to be terminated. The process number can be viewed using the “ps” command or the “jobs” command. A root user will affect the user's process, and a non-root user can only affect his or her own process. The general syntax of the kill command is as follows：

```
kill [ -s signal | -p ] [ -a ] pid ...
kill -l [ signal ]
```

Partial parameter are as follows：

- -s：Specifies the signal to send
- -p：simulate transmitting signal
- -l：Specifies the name list of signals
- pid：The ID number of the process to abort
- Signal：According to the signal

First use “ps -ef” and the pipe command to determine the PID to kill the process.

```
root@mys-8mmx:~# ps -ef | grep wpa_supplicant
root        564       1  0 Sep04 ?        00:00:00 /usr/sbin/wpa_supplicant
 -u
root        989     850  0 02:05 ttymxc1  00:00:00 grep wpa_supplicant
```

Then type the following command to terminate the process：

```
root@mys-8mmx:~# kill 564
```

The "killall" command terminates all processes within the same process group, allowing you to specify the name of the process to be terminated instead of the PID process number.

```
root@mys-8mmx:~#  killall  wpa_supplicant
root@mys-8mmx:~#
```

9. Application Development

This chapter mainly introduces some basic information for secondary development of the current SDK. The current SDK provides two configuration reference images, one is “myir-image-core”, which is mainly for non-GUI applications; the other is “myir-image-full”, which adds some GUI applications on the basis of “myir-image-core”, such as Qt runtime libraries, graphics processing related libraries and demo applications. For information about these two images, please refer to the “MYS-8MMX SDK2.0.0 Release notes”.

9.1. Development 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：

- ➢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)

The script executes as follows：

```
root@mys-8mmx:~# echo "echo 'myir test'" > shell_demo.sh
root@mys-8mmx:~# sh shell_demo.sh
myir test
root@mys-8mmx:~# bash shell_demo.sh
myir test
```

2) C/C++

C/C++ is the most commonly used programming language for low-level application development under the Linux platform, and is also the most efficient language after assembly. Development using C/C++ usually adopts the mode of cross-development, that is, development is carried out on the development host side, binary execution files are compiled and generated running on the target machine, and then deployed to run on the target machine.In this way, the SDK built based on Yocto needs to be installed first. Please refer to "MYS-8MMX Software Development Guide" for installation steps. After installation, the SDK environment needs to be configured as follows:

This section demonstrates application development by writing a simple Hello World instance, and the following demo program "hello.c" is written on the development host side:

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

Next, the application is compiled with "$CC", because the corresponding header file and link are needed when compiling. "$CC" contains the corresponding system library and configuration information. If you directly compile with "aarch64-poky-linux-gcc", you will not find the header files.

```
PC$ source ~/opt_meta_5_4/environment-setup-aarch64-poky-linux
PC$ cat main.c
#include <stdio.h>


int main(int argc, char *argv[])
{
        print("myir test!\n");
        return 0;
}
duxy@myir-server1:~/test$ $CC main.c -o main
duxy@myir-server1:~/test$ file main
main: ELF 64-bit LSB shared object, ARM aarch64, version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-aarch64.so.1, BuildID[sha1]=b24bfdeaefb3ca10ebf845be5dbc95bbd2aa5c1d, for GNU/Linux 3.14.0, not stripped
```

The generated execution file is then copied to the target machine through the "scp" command and executed as follows:

```
root@mys-8mmx:~#  ./main
myir test!
```

For more complex examples and development methods, refer to the "MYS-8MMX Software Development Guide" for instructions on application migration.

3) Python

Python is a high-level programming language with interpreted, object-oriented, dynamic data types. Python was invented by Guido van Rossum in late 1989, and the first public release was released in 1991. Like the Perl language, the Python source code is also licensed under the GPL(GNU General Public License). This section focuses on testing the use of Python, both from the Python command line and from the script.

- **Check the Supported Version of Python**

```
root@mys-8mmx:~# python
python             python2             python2.7            python3            python3.7m
python-config      python2-config      python2.7-config     python3.7          python3.7m-config-lib
```

- **Python Interactive Mode**

Start Python and enter the following text at the Python prompt, then press Enter to see how it works:

```
root@mys-8mmx:~#  python3
Python 3.7.5 (default, Sep  5 2020, 00:13:46)
```

```
[GCC 9.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print("myir test")
```

In Python version 3.7.5, the output from the above instance is as follows:

```
myir test
>>>
```

Run “exit()” function or press “ctrl+D” to exit Python:

```
>>> exit()
root@myir:~#
```

- **Write a Python Script for Testing**

Write a simple Python script, and all Python files will have the.py extension.

```
root@mys-8mmx:~#  vi test.py
root@mys-8mmx:~#  cat test.py
#!/usr/bin/env python3
print("myir test")
```

The Python3 interpreter executes the script in the */usr/bin* directory using the following command.

```
root@mys-8mmx:~#  chmod a+x test.py
root@mys-8mmx:~#  ./test.py
myir test
```

The Python3 interpreter is called with script parameters to start executing the script until it is finished. When the script has finished executing, the interpreter is no longer valid. The current system supports PIP and PIP3. When downloading a package, if you encounter authentication failure, you can proceed as follows:

```
pip install numpy -i http://pypi.douban.com/simple/ --trusted-host pypi.douba
n.com
```

9.2. Database

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

1) System SQLlite

SQLite is an embedded SQL database engine. Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to a normal disk file. A complete SQL database containing multiple tables, indexes, triggers, and views is contained in a single disk file. This is a lightweight database, an ACID-compliant relational database management system, designed for embedded use and currently used in many embedded products. Its footprint is very low, and in embedded devices, it may only need a few hundred K of memory. This database runs faster than MySQL or PostgreSQL, so here's a quick test.

- **Create a Database**

Start sqlite3 and create a new database <testdb.db>. Enter the following command into the terminal interface.

```
root@mys-8mmx:~#  sqlite3 testDB.db
SQLite version 3.29.0 2019-07-10 17:32:03
Enter ".help" for usage hints.
sqlite>
```

The above command creates 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 a “sqlite>” prompt after successfully creating the database file.

With the database created, you can use the ” .databases” command of SQLite to check if it is in the database list, as shown below:

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

Exit the SQLite prompt with the “.quit” command, as shown below:

```
sqlite>  .quit
root@mys-8mmx:~#
```

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

10. Reference

- **Linux kernel Open Source Community**

  https://www.kernel.org/

- **NXP development communities**

  https://community.nxp.com/

- **i.MX 8 Series Processor Introduction**

  https://www.nxp.com/products/processors-and-microcontrollers/arm-processors/i-mx-applications-processors/i-mx-8-processors:IMX8-SERIES

- **MCU SDK Resources Download**

  https://mcuxpresso.nxp.com/en/welcome

Appendix A

Warranty & Technical Support Services

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

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

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

Service Guarantee

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

Price

MYIR insists on providing customers with the most valuable products. We do not pursue excess profits which we think only for short-time cooperation. Instead, we hope to establish long-term cooperation and win-win business with customers. So we will offer reasonable prices in the hope of making the business greater with the customers together hand in hand.

Delivery Time

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

Technical Support

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

After-sale Service

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

Technical support service

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

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

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

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

After-sales maintenance service

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

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

Warm tips

1. MYIR does not supply maintenance service to LCD. We suggest the customer first check the LCD when receiving the goods. In case the LCD cannot run or no display, customer should contact MYIR within 7 business days from the moment get the goods.

2. Please do not use finger nails or hard sharp object to touch the surface of the LCD.

3. MYIR suggests user purchasing a piece of special wiper to wipe the LCD after long time use, please avoid clean the surface with fingers or hands to leave fingerprint.

4. Do not clean the surface of the screen with chemicals.

5. Please read through the product user manual before you using MYIR’ s products.

6. For any maintenance service, customers should communicate with MYIR to confirm the issue first. MYIR’ s support team will judge the failure to see if the goods need to be returned for repair service, we will issue you RMA number for return maintenance service after confirmation.

Maintenance period and charges

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

Shipping cost

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

Products Life Cycle

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

Value-added Services

1. MYIR provides services of driver development base on MYIR’ s products, like serial port, USB, Ethernet, LCD, etc.

2. MYIR provides the services of OS porting, BSP drivers’ development, API software development, etc.

3. MYIR provides other products supporting services like power adapter, LCD panel, etc.

4. ODM/OEM services.

MYIR Electronics Limited

Room 04, 6th Floor, Building No.2, Fada Road,

Yunli Inteiligent Park, Bantian, Longgang District.

Support Email: support@myirtech.com

Sales Email: sales@myirtech.com

Phone: +86-755-22984836

Fax: +86-755-25532724

Website: www.myirtech.com