软件应用指南

ECB30-P4T13IA5ME8G-I 评估板

成都亿佰特电子科技有限公司

Chengdu Ebyte Electronic Technology Co.,Ltd.

目录

免责申明和版权公告

本文中的信息，如有变更，恕不另行通知。 文档“按现状”提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档不负任何责任，包括使用本文档内信息产生的侵犯任何专利权行为的责任。本文档在此未以禁止反 言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

文中所得测试数据均为亿佰特实验室测试所得，实际结果可能略有差异。

文中提到的所有商标名称、商标和注册商标均属其各自所有者的财产，特此声明。

最终解释权归成都亿佰特电子科技有限公司所有。

注 意 ：

由于产品版本升级或其他原因，本手册内容有可能变更。亿佰特电子科技有限公司保留在没有任何通知或者提示的情况下对本手册的内容进行修改的权利。本手册仅作为使用指导，成都亿佰特电子科技有限公司尽全力在本手册中提供准确的信息，但是成都亿佰特电子科技有限公司并不确保手册内容完全没有错误，本手册中的所有陈述、信息和建议也不构成任何明示或暗示的担保。

1. 概述

Linux 软件评估指南用于介绍在亿佰特的开发板上运行开源 Linux 系统下的核心资源与外设资源的测试步骤与评估方法。本文可作为前期评估指导使用，也可以作为通用系统开发的测试指导书使用。

1.1. 软件资源

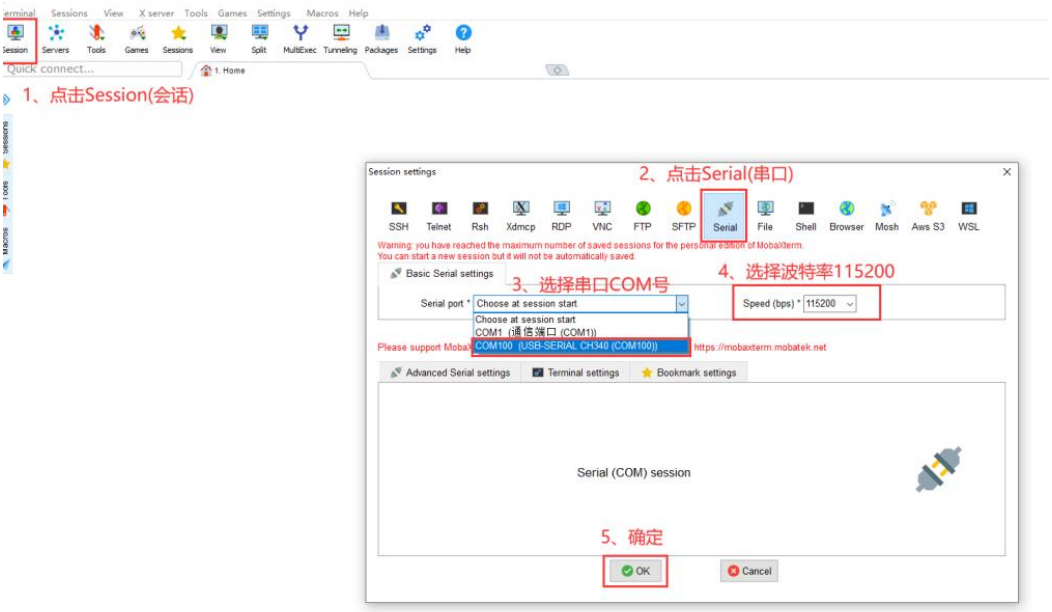
ECB30-P4T13IA5ME8G-I 搭载基于 Linux 5.4 版本内核的操作系统，开发板出厂附带嵌入式 Linux 系统开发所需要的交叉编译工具链， U-boot 源代码，Linux 内核和各驱动模块的源代码，以及适用于 Windows 桌面环境和 Linux 桌面环境的各种开发调试工具，应用开发样例等。

2. 使用前准备

2.1. 串口软件安装

使用串口前主要需要安装 CH340 串口驱动和 MobaXterm 串口终端。具体安装方法可见核心板开发指南。

连接好串口之后，安装 MobaXterm 后进行配置，按如下方式打开串口。



2.2. 快速启动

亿佰特出厂的单板机默认不带启动固件，需要用户自行烧录一个固件。用户可以根据单板机型号来选择烧录固件，这里我们主要帮助用户快速启动单板机，只讲解使用官方工具烧

录固件到 emmc 的方法，nand 也是一样的。更多烧录方法和烧录中的问题见 ECK30 的 Development_Guide。

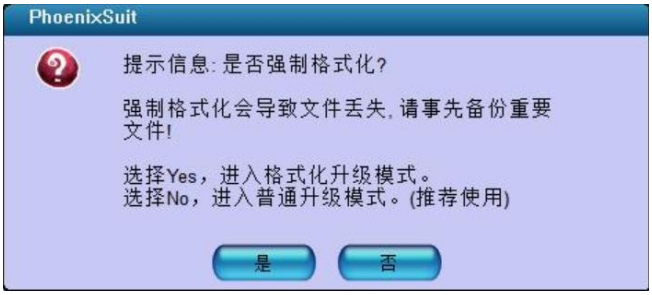
2.2.1. 烧录开发板 Flash

首先需要使用 usb 数据线连接单板机和电脑，接入单板机的 OTG 接口。

使用全志官方工具 PhoenixSuit 实现 USB 快速烧录。

点击上方的“一键刷机”项，进入到镜像选择。我们提供的镜像分为 nand 版本和 emmc 版本，用户需要选择正确的镜像。

用户不用管立即升级按钮， 只要在打开该软件， 选择好烧写镜像， usb 连线正常连接后，一旦待烧写小机进入烧写模式， 会自动识别并询问烧写情况：



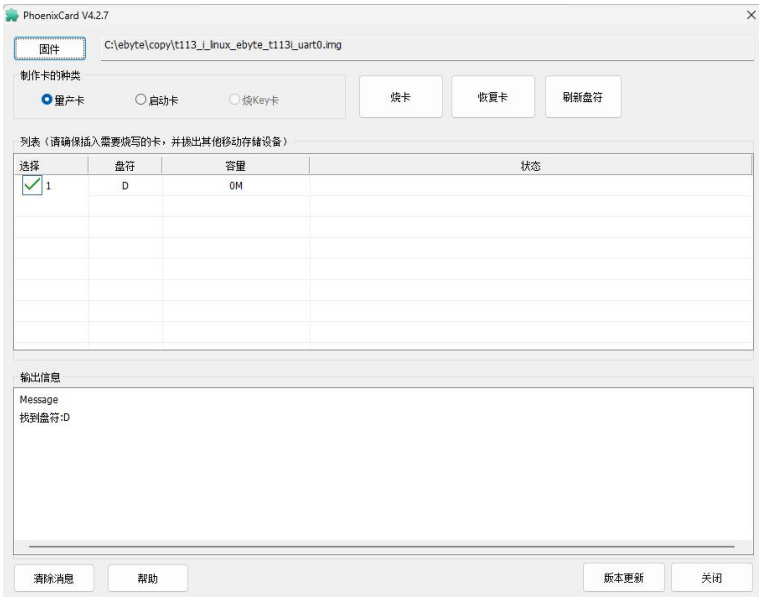
用户按需选择是否格式化升级，等待烧录完成即可。

2.2.2. 使用串口交互单板机

在烧录完成之后，我们可以接入串口来查看单板机的启动了。需要使用 usb 数据线连接单板机的调试串口。然后再使用上面的串口软件 mobaXterm 选择正确的串口号和波特率 115200 即可。接下来我们就可以进入第三章的功能测试内容了。

2.2.3. 烧录 SD 卡

这一节是对烧录功能的补充，用户可以烧录到 SD 卡来启动开发板，这里我们需要使用 PhoenixCard 来进行 SD 卡的烧录。



用户请依次选择待烧写固件，固件根据烧录的板子选择 emmc 版本的镜像即可， 比如手上的板子的 cpu 型号是 t113s4，那选择 t113_s4_linux_ebyte_t113s4_uart0.img。选择卡的种类为“启动卡” ， 而后选择待烧写的 SD 卡设备， 该软件会自动扫描可用的移动存储设备， 请注意不要选择了错误的设备。 最后点击“烧卡” 按钮， 烧卡过程中会实时显示当前进度。

3. 功能测试

3.1. 核心资源

在 Linux 系统中，提供了 proc 虚拟文件系统来查询各项核心资源的参数以及一些通用工具来评估资源的性能。下面将具体对 CPU，memory，eMMC，RTC 等核心资源的参数进行读取与测试。

3.1.1. CPU

3.1.1.1. 查看 CPU 信息

使用 cat /proc/cpuinfo 查看 CPU 信息。

```
#cat /proc/cpuinfo
processor       : 0
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 57.14
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

processor       : 1
model name      : ARMv7 Processor rev 5 (v7l)
BogoMIPS        : 57.14
Features        : half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae
CPU implementer : 0x41
CPU architecture: 7
CPU variant     : 0x0
CPU part        : 0xc07
CPU revision    : 5

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

processor：系统中逻辑处理核的编号，对于多核处理器则可以是物理核、或者使用超线程技术虚拟的逻辑核

model name：CPU 属于的名字及其编号

BogoMIPS：在系统内核启动时粗略测算的 CPU 每秒运行百万条指令数（MillionInstructions Per Second）

3.1.1.2. 查看 CPU 使用率

```
top - 13:56:36 up 58 min,  0 users,  load average: 0.00, 0.00, 0.11
Tasks:  64 total,   1 running,  33 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  1.0 sy,  0.0 ni, 99.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   449616 total,   370472 free,    30388 used,    48756 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   407640 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  255 root      20   0    4908   2284   1868 R  0.7  0.5   0:00.16 top
    1 root      20   0   28520   5696   4472 S  0.0  1.3   0:03.30 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kthreadd
    3 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_par_gp
    5 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 slub_flushwq
    6 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 netns
    8 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/0:0+
    9 root      20   0       0      0      0 I  0.0  0.0   0:03.06 kworker/u2:+
   10 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 mm_percpu_wq
   11 root      20   0       0      0      0 I  0.0  0.0   0:00.00 rcu_tasks_k+
   12 root      20   0       0      0      0 I  0.0  0.0   0:00.00 rcu_tasks_t+
   13 root      20   0       0      0      0 S  0.0  0.0   0:00.25 ksoftirqd/0
   14 root      20   0       0      0      0 I  0.0  0.0   0:00.34 rcu_preempt
   15 root      20   0       0      0      0 S  0.0  0.0   0:00.02 kdevtmpfs
   16 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 inet_frag_wq
   20 root      20   0       0      0      0 I  0.0  0.0   0:00.77 kworker/0:4+
```

%usr：表示用户空间程序的 cpu 使用率

%sys：表示系统空间的 cpu 使用率

%idle：空闲 cpu

%irq：cpu 处理硬中断的数量

%sirq：cpu 处理软中断的数量

3.1.1.3. 查看 CPU 温度信息

通过 cat /sys/class/thermal/thermal_zone0/temp 来查看 CPU 的内部温度。

```
#cat /sys/class/thermal/thermal_zone0/temp
39061
```

上面的数值除以 1000 就是正确的温度值，单位为摄氏度。

3.1.2. 内存

3.1.2.1. 查看内存信息

通过 cat /proc/meminfo 可以查看内存信息。

```
#cat /proc/meminfo
MemTotal:         231080 kB
MemFree:          197528 kB
MemAvailable:     206076 kB
Buffers:            3596 kB
Cached:             8052 kB
SwapCached:            0 kB
Active:             9908 kB
```

MemTotal: 总内存量。这里显示的值为 449616 kB，表示系统总共有大约 449 MB 的物理内存可用。

MemFree: 空闲内存量。这里显示的值为 370472 kB，表示当前系统中有大约 370 MB 的内存是空闲的，未被使用。

MemAvailable: 可用内存量。这里显示的值为 407640 kB，表示当前可供系统使用的内存总量，包括已缓存的内存和可用的内存。

Buffers: 缓冲区使用量。这里显示的值为 0 kB，表示系统当前没有使用任何内存作为缓冲区。

Cached: 缓存的内存量。这里显示的值为 39208 kB，表示系统当前用于缓存的内存量。

SwapCached: 交换缓存的内存量。这里显示的值为 0 kB，表示当前没有被缓存到交换空间中的内存。

Active: 活跃的内存量。这里显示的值为 13408 kB，表示当前正在使用的内存量。

3.1.2.2. 内存压力测试

通过给定测试内存的大小和次数, 可以对系统现有的内存进行压力上的测试。可使用系统工具 memtester 进行测试，如指定内存大小 10MB，测试次数为 1，测试命令为“memtester 10M 10"。

```
#memtester 10M 1
memtester version 4.3.0 (32-bit)
Copyright (C) 2001-2012 Charles Cazabon.
Licensed under the GNU General Public License version 2 (only).

pagesize is 4096
pagesizemask is 0xfffff000
want 10MB (10485760 bytes)
got  10MB (10485760 bytes), trying mlock ...locked.
Loop 1/1:
  Stuck Address       : ok
  Random Value        : ok
  Compare XOR         : ok
  Compare SUB         : ok
  Compare MUL         : ok
  Compare DIV         : ok
  Compare OR          : ok
  Compare AND         : ok
  Sequential Increment: ok
  Solid Bits          : ok
  Block Sequential    : ok
  Checkerboard        : ok
  Bit Spread          : ok
  Bit Flip            : ok
  Walking Ones        : ok
  Walking Zeroes      : ok
  8-bit Writes        : ok
  16-bit Writes       : ok

Done.
```

3.1.3. eMMC 测试

查看 emmc 容量

通过 fdisk-l 命令可以查询到 eMMC 分区信息及容量。

```
#fdisk -l
Found valid GPT with protective MBR; using GPT

Disk /dev/mmcblk0: 15147008 sectors, 3300M
Logical sector size: 512
Disk identifier (GUID): ab6f3888-569a-4926-9668-80941dcb40bc
Partition table holds up to 7 entries
First usable sector is 73728, last usable sector is 15146974

Number  Start (sector)    End (sector)  Size Name
     1           73728          108165 16.8M boot-resource
     2          108166          110213 1024K env
     3          110214          112261 1024K env-redund
     4          112262          147461 17.1M boot
     5          147462         2244613 1024M rootfs
     6         2244614         2277381 16.0M private
     7         2277382        15146974 6283M UDISK
Disk /dev/sda: 29 GB, 30784094208 bytes, 60125184 sectors
3742 cylinders, 255 heads, 63 sectors/track
Units: sectors of 1 * 512 = 512 bytes

Device  Boot StartCHS    EndCHS        StartLBA     EndLBA    Sectors  Size
Id Type
/dev/sda1       0,0,33          1023,254,63              32    60125183    60125152 28.6G    c
Win95 FAT32 (LBA)
```

查看 emmc 分区

通过 df 命令可以查询到 eMMC 分区信息，使用情况，挂载目录等信息。

```
#df -h
Filesystem                Size      Used Available Use% Mounted on
/dev/root               991.9M    180.5M    795.4M  18% /
devtmpfs                484.5M         0    484.5M   0% /dev
tmpfs                   493.7M         0    493.7M   0% /dev/shm
tmpfs                   493.7M    204.0K    493.5M   0% /tmp
tmpfs                   493.7M    128.0K    493.5M   0% /run
/dev/sda1                28.7G    269.5M     28.4G   1% /mnt/usb/sda1
/dev/by-name/UDISK        6.1G      4.0K      6.1G   0% /mnt/UDISK
```

3.1.4. NandFlash 测试

3.1.4.1. 查看 Nand 容量及大小

```
#cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00100000 00040000 "boot0"
mtd1: 00300000 00040000 "uboot"
mtd2: 00100000 00040000 "secure_storage"
mtd3: 0fb00000 00040000 "sys"
```

3.1.4.2. Nand 的性能测试

写文件

性能测试主要测试 nand 在 linux 系统下对文件的读写速度，一般结合 time 与 dd 双命令进行测试。

```
#time dd if=/dev/zero of=write_file bs=100M count=1 conv=fsync
100+0 records in
100+0 records out

real    0m16.041s
user    0m0.000s
sys     0m2.115s
```

通过计算得知写文件的速度约为 6.25MB/s。

读文件

在嵌入式系统中，经常需要测试系统文件读写性能，读文件时忽略 cache 的影响。先执行下下面的命令清除 cache。

```
#sync; echo 3 > /proc/sys/vm/drop_caches
#time dd if=write_file of=read_file bs=100M count=1
1+0 records in
1+0 records out

real    0m15.925s
user    0m0.000s
sys     0m0.937s
```

3.1.5. RTC

Linux 系统中分系统时钟（软件时钟）和 RTC 时钟（硬件时钟），系统时钟掉电即会消失， RTC 时钟在安装电池的情况下会长期运行。如需使用外部 RTC 时钟， 请将 ML2032（3V 可充）或 CR2032（3V 不可充） 电池安装至 RTC 纽扣电池座。

设置系统时间

将系统时间设置为 2024.07.12 16:35:15

```
# date 071216352024.15
Fri Jul 12 16:35:15 UTC 2024
```

将系统时间写入 RTC

```
# hwclock -w
```

```
# hwclock -r
2000-01-01 07:07:15.965766+0000
```

掉电保持 RTC 时间

将开发板关机断开电源，一段时间后重新上电开机。查看 RTC 时间和系统时间：

```
# hwclock -r
2024-07-12 16:52:42.648561+0000
```

3.1.6. WatchDog

使用测试例程测试看门狗，文件已经被编译好放在 tool 目录下，拷贝到开发板后。运行看门狗应用,超时时间为 4s，每间隔 1s 喂一次狗：

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

如果将上面的 1s 改到大于 4s, 则超过了要求的 4s 喂狗时间，开发板会重启。

3.1.7. 电源管理

本章节演示 Linux 电源管理的 Suspend 功能，让开发板睡眠 ，通过外部事件唤醒。Linux 内核一般提供了三种 Suspend: Freeze 和 STR(Suspend to RAM)，在用户空间向”/sys/power/state”文件分别写入”freeze”“和”mem”，即可触发它们。

查看当前支持的模式

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

休眠到内存

```
root@ebyte-ubuntu:~# echo "mem" > /sys/power/state
```

3.2. 外设接口

3.2.1. GPIO

查看所有 GPIO，列出所有 registered 的 pins。

```
#mount -t debugfs none /sys/kernel/debug
#cat /sys/kernel/debug/pinctrl/pio/pins
registered pins: 88
pin 32 (PB0)
pin 33 (PB1)
pin 34 (PB2)
pin 35 (PB3)
pin 36 (PB4)
pin 37 (PB5)
pin 38 (PB6)
pin 39 (PB7)
pin 40 (PB8)
pin 41 (PB9)
pin 42 (PB10)
pin 43 (PB11)
pin 44 (PB12)
pin 64 (PC0)
pin 65 (PC1)
******
```

导出 GPIO

```
#cd /sys/class/gpio/
#ls
export        gpiochip0    unexport
#echo 32 > export
#ls
export        gpio32        gpiochip0    unexport
```

设置 GPIO 方向

输入

```
#cd gpio32
#ls
active_low                edge                    uevent
device                    power                   value
direction                 subsystem               waiting_for_supplier
#echo in > direction
```

输出

```
#echo out > direction
```

设置 GPIO 值为 1 之后可以使用万用表进行测量，可以测量到 J10 的 31 号脚为 3.3V。

```
#echo 1 > value
```

注销 GPIO

```
#echo 32 > unexport
]#ls
export        gpiochip0    unexport
```

3.2.2. LED 灯

Linux 系统提供了一个独立的子系统以方便从用户空间操作 LED 设备，该子系统以文件的形式为 LED 设备提供操作接口。这些接口位于/sys/class/leds 目录下。在硬件资源列表中，我们已经列出了开发板上所有的 LED。下面通过命令读写 sysfs 的方式对 LED 进行测试。下述命令均为通用命令，也是操控 LED 的通用方法。

```
#cd /sys/class/leds/
#ls
heartbeat
```

点亮和熄灭 LED

```
# echo 0 > brightness
# echo 1 > brightness
```

查看 LED 触发模式

```
root@ebyte-ubuntu:/sys/class/leds/green:heartbeat# cat trigger
[none] rfkill-any rfkill-none kbd-scrolllock kbd-numlock kbd-capslock kbd-kanalock kbd-shiftlock kbd-altgrlock kbd-ctrllock kbd-altlock kbd-shiftllock kbd-shiftrlock kbd-ctrlllock kbd-ctrlrlock timer oneshot heartbeat backlight gpio cpu cpu0 default-on transient flash torch mmc0 mmc1
```

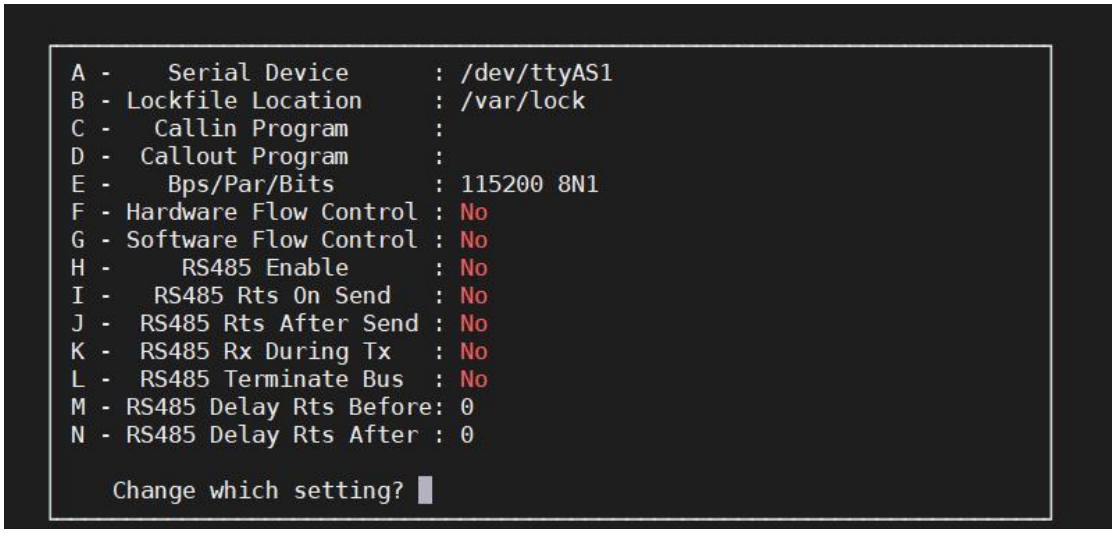
3.2.3. 串口

本单板机引出了两路串口供使用，分别是 UART0（DEBUG）和 UART1。

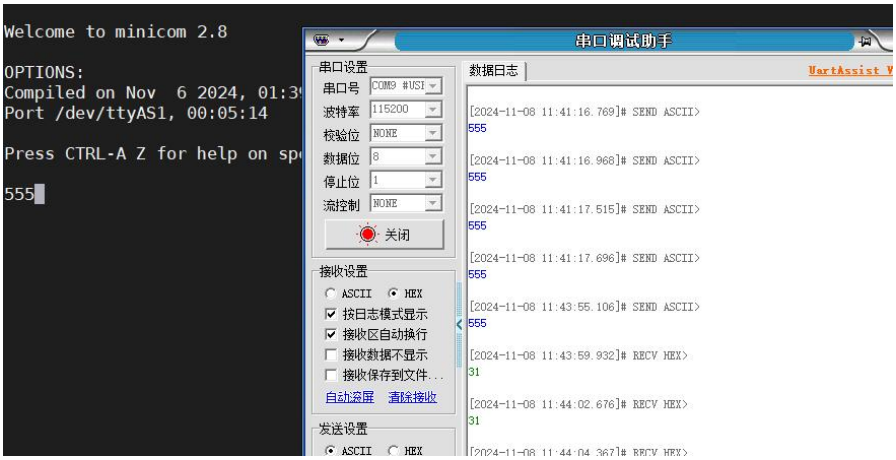
串口 1 需要使用设备树配置进行打开

```
&uart1 {
 pinctrl-names = "default", "sleep";
 pinctrl-0 = <&uart1_pins_a>;
 pinctrl-1 = <&uart1_pins_b>;
 status = "okay";
};

 uart1_pins_a: uart1_pins@0 {
  pins = "PB8", "PB9";
  function = "uart1";
  drive-strength = <10>;
  bias-pull-up;
 };

 uart1_pins_b: uart1_pins@1 {
  pins = "PB8", "PB9";
  function = "gpio_in";
 };
```

此串口可以直接使用 usb-ttl 转接线连接 PC 进行测试。在单板机中被注册为 ttyAS1，使用 minicom 打开 ttyAS1 即可通讯。



可以使用 minicom 进行正常收发数据



3.2.4. USB

本节通过相关命令或热插拔、USB HUB 验证 USB Host 驱动的可行性，实现读写 U 盘的功能、usb 枚举功能。

3.2.4.1. 1.插入 U 盘

插入 U 盘和使用 df-h 查找存储设备。

```
#df -h
Filesystem                Size      Used Available Use% Mounted on
ubi0_5                   83.1M     41.2M     41.9M  50% /
tmpfs                   113.3M    100.0K    113.2M   0% /tmp
tmpfs                   113.3M     16.0K    113.3M   0% /run
devtmpfs                104.8M         0    104.8M   0% /dev
/dev/by-name/UDISK        2.8M     28.0K      2.5M   1% /mnt/UDISK
/dev/sda1                28.7G    326.1M     28.3G   1% /mnt/usb/sda1
```

3.2.4.2. U 盘挂载读写

挂载 U 盘,系统会自动挂载 u 盘，/dev/sda1 到/mnt/usb/sda1/目录下

读文件

提前在 u 盘中创建一些文件

```
# cd /mnt/usb/sda1/
#cat mic_in_test.sh
tinymix set "ADC1 Input FMINL Switch" 0
tinymix set "ADC1 Input MIC1 Boost Switch" 1
tinymix set "ADC1 Input LINEINL Switch" 0
***
```

写文件

```
#touch test
#echo hello > test
#cat test
hello
```

3.2.5. OTG

3.2.5.1. USB HOST 模式测试

请通过 Type-C 转接头，将 U 盘与评估板 USB0 DRD 接口连接。

系统上电启动， USB0 DRD 接口默认为 DEVICE 模式， 当接入 DEVICE 设备时，系统将自动切换为 HOST 模式。执行如下命令，可查看到当前为 HOST 模式。

```
#cat /sys/bus/platform/drivers/otg\ manager/soc\@3000000\:usbc0\@0/otg_role
usb_host
```

3.2.5.2. USB DEVICE 模式测试

系统上电启动， USB0 DRD 接口默认配置为 DEVICE 模式，当接入 DEVICE 设备时，系统将自动切换为 HOST 模式。请确保 USB0 DRD 接口未连接设备，然后执行如下命令， 可查看到当前为 DEVICE 模式。

```
#cat /sys/bus/platform/drivers/otg\ manager/soc\@3000000\:usbc0\@0/otg_role
usb_device
```

将产品资料“4-软件资料\Demo\base-demos\otg\”目录下的 otg.sh 脚本文件拷贝至评估板文件系统"/opt/"目录下。 执行如下命令， 将评估板 10MByte 容量的 DDR 内存空间虚拟为 U 盘。

在挂载 U 盘之前，需要将开机启动程序 S50adb_start 移动到其他目录，否则脚本将执行不成功。

```
#mv S50adb_start /root/
```

然后执行脚本 otg.sh

```
#!/bin/bash
dd if=/dev/zero of=/dev/a.bin bs=1M count=10
mount -t configfs none /sys/kernel/config
mkdir /sys/kernel/config/usb_gadget/g1
echo "0x18d1" > /sys/kernel/config/usb_gadget/g1/idVendor
echo "0x0001" > /sys/kernel/config/usb_gadget/g1/idProduct
mkdir /sys/kernel/config/usb_gadget/g1/strings/0x409
mkdir /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0
echo /dev/a.bin > /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/lun.0/file
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1
echo 0xc0 > /sys/kernel/config/usb_gadget/g1/configs/c.1/bmAttributes
echo 500 > /sys/kernel/config/usb_gadget/g1/configs/c.1/MaxPower
mkdir /sys/kernel/config/usb_gadget/g1/configs/c.1/strings/0x409
ln -s /sys/kernel/config/usb_gadget/g1/functions/mass_storage.usb0/
/sys/kernel/config/usb_gadget/g1/configs/c.1/
ls /sys/class/udc/ | xargs echo > /sys/kernel/config/usb_gadget/g1/UDC
```

使用 typec 数据线连接到电脑，可以看到需要格式化的信息，格式化完成后将会有一个 10M 的分区。

3.2.6. SD 卡

本小节使用 SanDisk 公司、 32GB 容量的 Micro SD 卡来测试评估板 Micro SD 接口性能。请参考《Linux 系统启动卡制作及系统固化》文档将其制作成 Linux 系统启动卡，再进行测试。不同的 Micro SD 卡以及不同的测试方法，对 Micro SD 接口测试结果将造成一定差异。

请将 Linux 系统启动卡插至评估板 Micro SD 卡槽，评估板上电，进入评估板文件系统执行如下命令查看 Linux 系统启动卡信息。

3.2.6.1. 查看 TF 卡容量

通过 fdisk -l 命令可以查询到 TF 卡分区信息及容量：

```
#fdisk -l
Found valid GPT with protective MBR; using GPT
Disk /dev/mmcblk0: 62333952 sectors, 1764M
Logical sector size: 512
Disk identifier (GUID): ab6f3888-569a-4926-9668-80941dcb40bc
Partition table holds up to 12 entries
First usable sector is 73728, last usable sector is 62333947
Number  Start (sector)    End (sector)  Size Name
     1           73728          108165 16.8M boot-resource
     2          108166          110213 1024K env
     3          110214          112261 1024K env-redund
     4          112262          147461 17.1M boot
     5          147462         2244613 1024M rootfs
     6         2244614         2246661 1024K dsp0
     7         2246662         2248709 1024K riscv0
     8         2248710         2281477 16.0M private
     9         2281478        62331903 28.6G UDISK
```

3.2.6.2. TF 卡的性能测试

性能测试主要测试 eMMC 在 linux 系统下对文件的读写速度，一般结合 time 与 dd 双命令进行测试。挂载需要测试的 TF 卡分区，这里以最后一个分区/dev/mmcblk0p9 为例，挂载目录为/run/meida/mmcblk0p9。

执行如下命令，创建目录并对分区进行挂载。

```
mkdir -p /run/media/mmcblk0p9
mount /dev/mmcblk0p9 /run/media/mmcblk0p9
```

Micro SD 接口写速度测试

进入评估板系统， 执行如下命令测试 Micro SD 接口写速度

```
echo 3 > /proc/sys/vm/drop_caches
time dd if=/dev/zero of=/run/media/mmcblk0p9/test bs=1024K count=100 conv=fsync


100+0 records in
100+0 records out


real    0m6.473s
user    0m0.000s
sys     0m1.633s
```

time 命令有计时作用， dd 用于复制，从 if(input file)文件读出，写到 of(output file)指定的文件， bs 是每次写块的大小， count 是读写块的数量。

"if=/dev/zero"不产生 IO，即不断输出数据，可用来测试纯写速度。

此处一共写 100MByte 测试数据至 Linux 系统启动卡的 test 文件，可看到本次测试的 Micro SD 接口写速度约为： 100MByte / 6.47s ≈ 15.45MB/s。

Micro SD 接口读速度测试

执行如下命令测试 Micro SD 接口读速度。

```
echo 3 > /proc/sys/vm/drop_caches
time dd if=/run/media/mmcblk0p9/test of=/dev/null bs=1024K count=100
```

"of=/dev/null"不产生 IO，即不断接收数据，可用来测试纯读速度。

此处从 test 文件一共读出 100MByte 的数据，可看到本次测试的 Micro SD 接口读速度约为： 100MByte / 4.47s ≈ 22.40MB/s。

测试完成后， 执行如下命令卸载挂载分区。

```
umount /run/media/mmcblk0p9/
rm -r /run/media/mmcblk0p9
```

3.2.7. 显示

本模块由显示引擎（DE）和各类型控制器（tcon）组成。输入图层（layers）在 DE 中进行显示相关处理后，通过一种或多种接口输出到显示设备上显示，以达到将众多应用渲染的 图层合成后在显示器呈现给用户观看的作用。 DE 有 2 个独立单元（可以简称 de0、de1），可以 分别接受用户输入的图层进行合成，输出到不同的显示器，以实现双显。 DE 的每个独立的单元有 1-4 个通道（典型地， de0 有 4 上， de1 有 2 个），每个通道可以同时处理接受 4 个格式相同的 图层。 sunxi 平台有视频通道和 UI 通道之分。视频通道功能强大，可以支持 YUV 格式和 RGB 图层。 UI 通道只支持 RGB 图层。

3.2.7.1. 5.5 英寸 MIPI 显示

请将 5.5 英寸 MIPI 显示屏（型号： 亿佰特 ECA11-5P5LCDMIPI1019CT-C，分辨率：1080x1920）与评估板的 MIPILCD（显示）、 CAP TS（触摸） 接口正确连接。

注意： 请务必使用我司配套的 7 英寸 MIPI 显示屏、 FFC 软排线，并按照如下方法进行硬件连接。 若采用第三方配件，需仔细核对评估板接口、 FFC 软排线、 MIPI 显示屏三者线序， 否则可能烧毁 MIPI LCD 屏。

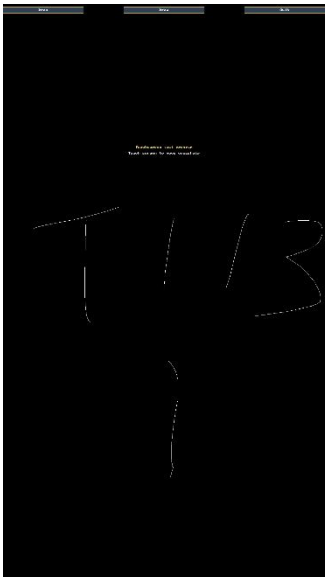
请将 26pin FFC 软排线连接 MIPI LCD 屏至评估板 MIPI LCD 接口。

评估板重启后， MIPI 显示屏可观察到启动 logo。

Emmc 板支持 QT，而 Nand 板不支持 QT，nand 板用户可以通过 ts_test 命令来使用显示和触摸。

也可以使用截屏的方式来调试。这里使用 ts_test 绘出 T113i 的图像之后并截屏，同时验证了显示和触摸。

```
echo 0 > /sys/class/disp/disp/attr/disp
echo filename.bmp > /sys/class/disp/disp/attr/capture_dump
```

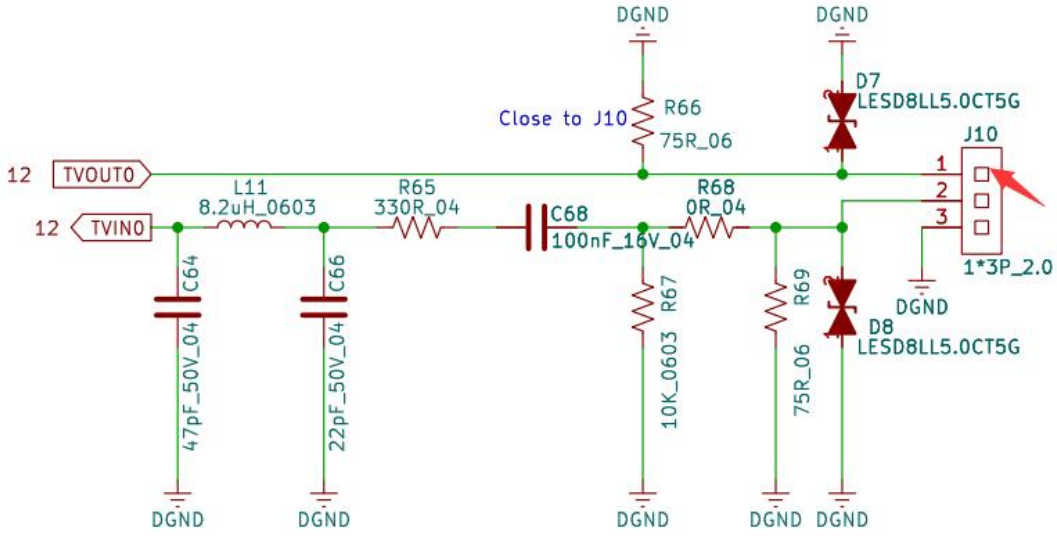


3.2.7.2. CVBS 显示

CVBS 输出

Cvbs 显示需要改设备树文件使得 screen0_output_type 和 screen0_output_mode 切换到 cvbs 模式。

在编译好 kernel 和 uboot 之后，替换板上的镜像文件，使用显示器连接好 cvbs out 引脚。显示器可以正常显示图像。

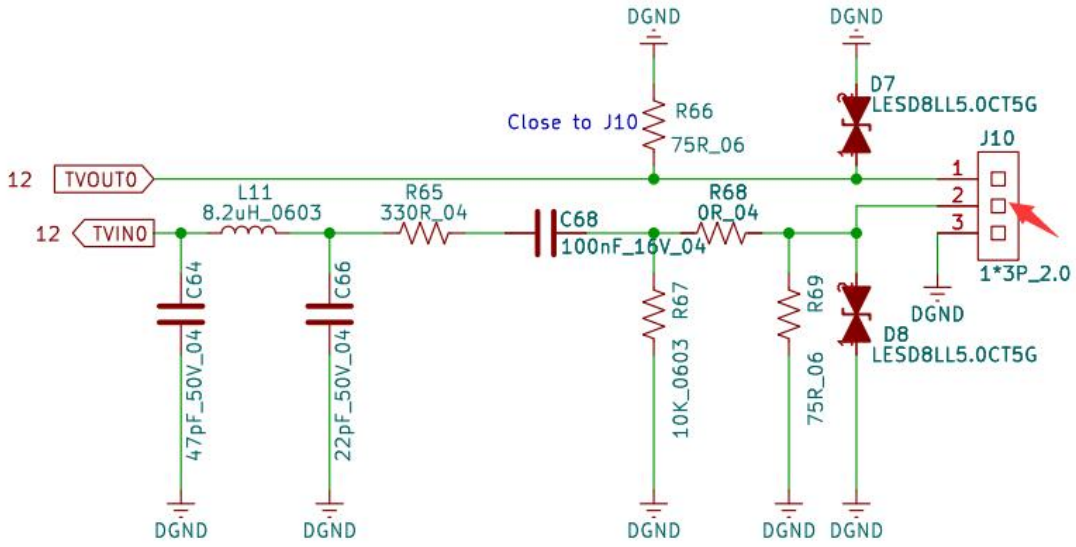


模拟信号接口

CVBS 输入

我们使用摄像头来作为 cvbs 的信号输入，也可以采用 cvbsout 和 cvbsin 短接的方法来验证输入信号。

将摄像头连接到 CVBS in 引脚上



模拟信号接口

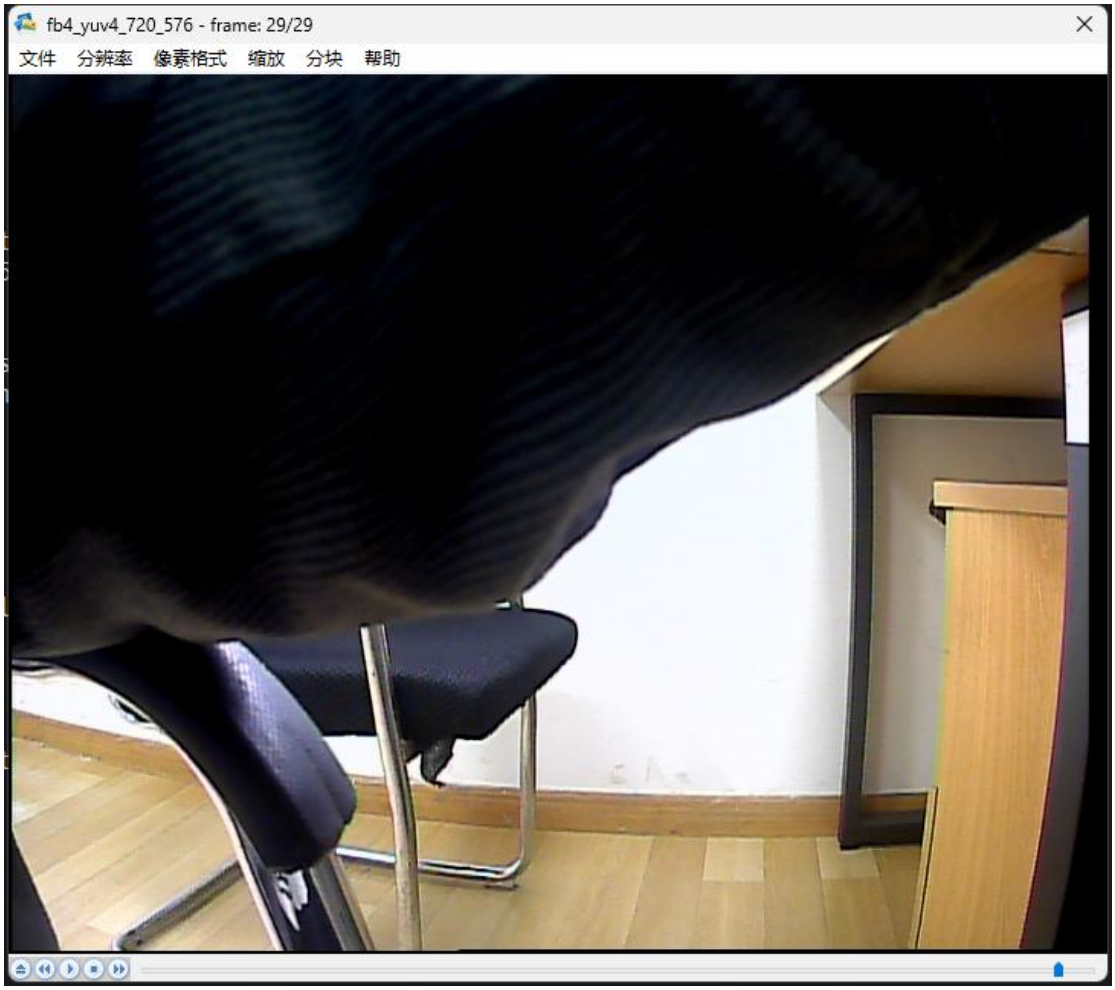
为摄像头上电之后，使用 tvd_test_mmap 来采集摄像头的输入。我们需要根据摄像头的宽高来指定这里的参数，其他参数使用默认配置即可。然后会在 root 目录生成一个 fb4_yuv4_720_576.bin。

```
tvd_test_mmap 4 0 720 576 /root 4 30 30
#tvd_test_mmap -h
============================Usage============================
tvd_test_mmap <videoX> <sel> <width> <height> <path> <format_mode> <test_cnt> <fps>
videoX                means: open /dev/videoX
sel                   means: select subdev-X
width                 means: camera capture pic width
height                means: camera capture pic height
path                  means: camera capture save file path, default "/tmp/tvd_test/"
format_mode           means: camera capture pic format, default 4(NV21)
test_cnt              means: camera capture pic count, default 20
fps                   means: camera capture fps, default 30
============================Usage============================
```

我们需要将这个 bin 文件拷到 windows 用特定的软件来打开，这里使用 U 盘来传输。

```
#cp fb4_yuv4_720_576.bin /mnt/usb/sda1/
```

使用 YUV Player 来打开，需要设置分辨率为 720*576，设置像素格式为 NV21，然后打开 fb4_yuv4_720_576.bin 即可。



3.3. 网络接口

3.3.1. Ethernet

使用 net-tools 工具包中的 ifconfig 对网络进行手动配置，首先通过通过 ifconfig 命令查看网络设备信息如下。

```
#ifconfig eth0 192.168.1.250 netmask 255.255.255.0 up
#ifconfig eth0
eth0        Link encap:Ethernet  HWaddr 2E:3A:BA:FB:7A:B6
            inet addr:192.168.1.250  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::2c3a:baff:fefb:7ab6/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:586 (586.0 B)
            Interrupt:37
```

配置好以太网连接之后就可以使用 PING 对网络连接进行简单的测试，ping 同一网段的虚拟机测试。

```
ping 192.168.1.130
PING 192.168.1.130 (192.168.1.130) 56(84) bytes of data.
64 bytes from 192.168.1.130: icmp_seq=1 ttl=64 time=1.53 ms
64 bytes from 192.168.1.130: icmp_seq=2 ttl=64 time=1.34 ms
64 bytes from 192.168.1.130: icmp_seq=3 ttl=64 time=1.37 ms
64 bytes from 192.168.1.130: icmp_seq=4 ttl=64 time=1.30 ms
```

3.3.1.1. Iperf3

iperf3 是在 IP 网络上主动测量最大可实现带宽的工具。 它支持调节测试时间、缓冲区大小和协议(IPV4 和 IPV6 下的 TCP、 UDP、 SCTP)等各种参数。 iperf3 按角色可以分 为服务端模式或客户端模式，我们可以用它来测试和查看 TCP 模式下的网络带宽，TCP 窗口值，重传的概率等，也可以测试指定 UDP 带宽下丢包率，延迟和抖动情况。

我们在开发主机上打开 MobaXterm，带千兆网卡的主机作为 iperf3 的服务端，被测试的开发板作为客户端分别测试开发板网卡 TCP 和 UDP 的性能。

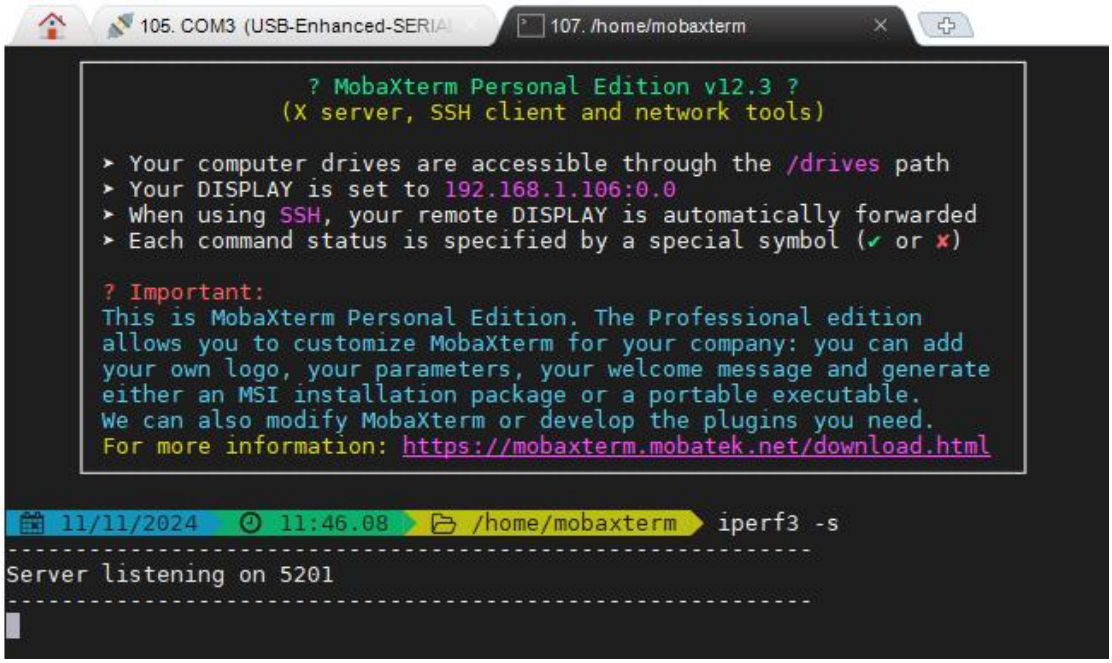
将服务器和开发板通过 CAT6 网线直连，并配置好各自的 IP 地址。例如我们设置服务器 ip 为 192.168.1.106，设开发板 IP 为 192.168.1.250, 并使用 ping 命令测试确保它

们之间是连通的。

注意：尽量不要连接路由器或交换机，以免测试结果受到中间设备传输转发的影响。

测试 TCP 性能

服务器上 iperf3 使用-s 参数表示工作在服务端模式。



开发板上运行的 iperf3 程序工作在客户端， TCP 模式，其中参数说明如下：

➢ -c 192.168.0.102 : 工作在客户端，连接服务端 192.168.1.106

➢ -i 2 : 测试结果报告时间间隔为 2 秒

➢ -t 10 : 总测试时长为 10 秒

```
#iperf3 -c 192.168.1.106 -i 2 -t 10
Connecting to host 192.168.1.106, port 5201
[  5] local 192.168.1.250 port 58936 connected to 192.168.1.106 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-2.00   sec   215 MBytes   900 Mbits/sec    0    754 KBytes
[  5]   2.00-4.00   sec   216 MBytes   906 Mbits/sec    0    754 KBytes
[  5]   4.00-6.00   sec   202 MBytes   851 Mbits/sec    0    847 KBytes
[  5]   6.00-8.00   sec   198 MBytes   828 Mbits/sec    0    936 KBytes
[  5]   8.00-10.00  sec   216 MBytes   907 Mbits/sec    0    936 KBytes
- - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  1.02 GBytes   878 Mbits/sec    0             sender
[  5]   0.00-10.00  sec  1.02 GBytes   876 Mbits/sec                  receiver

iperf Done.
```

客户端经过 10 秒之后测试结束并显示上面的测试结果，表明 TCP 带宽为 900 Mbits 左右。

测试 UDP 性能

服务器上继续运行 iperf3 使用-s 参数表示工作在服务端模式。

设备上 iperf3 工作在客户端， UDP 模式，其中参数说明如下：

- -u : 工作在 UDP 模式
- -c 192.168.1.106 : 工作在客户端，连接服务端 192.168.0.106
- -i 2 : 测试结果报告时间间隔为 2 秒
- -t 10 : 总测试时长为 10 秒
- -b 100M : 设定 UDP 传输带宽为 100Mbps.

```
#iperf3 -c 192.168.1.106 -u -i 2 -t 10 -b 100M
Connecting to host 192.168.1.106, port 5201
[  5] local 192.168.1.250 port 58311 connected to 192.168.1.106 port 5201
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  5]   0.00-2.00   sec  1.50 MBytes  6.28 Mbits/sec  1085
[  5]   2.00-4.00   sec  80.6 KBytes  330 Kbits/sec   57
[  5]   4.00-6.00   sec   322 KBytes  1.32 Mbits/sec  228
[  5]   6.00-8.00   sec   161 KBytes   660 Kbits/sec  114
[  5]   8.00-10.00  sec   242 KBytes   990 Kbits/sec  171
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  5]   0.00-10.00  sec  2.29 MBytes  1.92 Mbits/sec  0.000 ms  0/1655 (0%)  sender
[  5]   0.00-10.00  sec  2.29 MBytes  1.92 Mbits/sec  0.099 ms  0/1655 (0%)
receiver
```

客户端经过 10 秒之后测试结束并显示上面的测试结果，表明 UDP 在指定带宽为 100Mbps 时没有丢包。

3.3.2. WIFI

本节主要介绍 Linux 下 Wi-Fi 的配置和使用，通常 Wi-Fi 模块可以支持两种工作模式，分别是 STA 模式和 AP 模式，有些设备还支持 STA 和 AP 模式同时工作。 STA 模式允许设备连接外部 Wi-Fi 热点， AP 模式将设备变成 Wi-Fi 热点，供其它设备连接。

```
#ifconfig wlan0 up
```

Wi-Fi 的网络节点 wlan0, 如下所示

```
#ifconfig wlan0
wlan0     Link encap:Ethernet  HWaddr E0:3C:7A:F1:DC:40
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

3.3.2.1. STA 模式连接 WIFI 热点

扫描附近的 wifi 热点，得到附近 Wi-Fi 热点列表如下：

```
#iw dev wlan0 scan | grep SSID
        SSID: ChinaNet-ShJR
        SSID: TP-LINK_8BF6
        SSID: zhendu
        SSID: TP-LINK_\xef\xbc\x88\xe7\xa1\xac\xe4\xbb\xb6\xe7\xbb\x84)
        SSID: TEST_ZW
        SSID: xiaomi_gywl_yzd
                * SSID List
        SSID:
                * SSID List
```

下面尝试手动连接附近的 Wi-Fi 热点“TP-LINK_8BF6”，这是一个采用 WPA2 加密方式的 Wi-Fi 热点，密码为 EC12345678。

```
#iw dev wlan0 scan | grep SSID
        SSID: ChinaNet-ShJR
        SSID: TP-LINK_8BF6
        SSID: zhendu
        SSID: TP-LINK_\xef\xbc\x88\xe7\xa1\xac\xe4\xbb\xb6\xe7\xbb\x84)
        SSID: TEST_ZW
```

使用 wifi_set.sh 来进行 wifi 连接。目录在 tool 下。

```
#./wifi_setup.sh -i TP-LINK_8BF6 -p EC12345678
conf    : /etc/wpa_supplicant.conf
ssid    : TP-LINK_8BF6
psk     : EC12345678
device: wlan0
update /etc/wpa_supplicant.conf
startup wpa_supplicant ...
Successfully initialized wpa_supplicant
udhcpc: started, v1.33.2
udhcpc: sending discover
udhcpc: sending discover
udhcpc: sending select for 192.168.1.199
udhcpc: sending select for 192.168.1.199
udhcpc: lease of 192.168.1.199 obtained, lease time 7200
deleting routers
adding dns 114.114.114.114
adding dns 61.139.2.69
wifi setup successfully!
```

查看连接状态

```
#iw wlan0 link
Connected to ec:26:ca:6a:8b:f6 (on wlan0)
        SSID: TP-LINK_8BF6
        freq: 2412
        signal: -60 dBm
        tx bitrate: 150.0 MBit/s
```

网络连通测试，执行如下命令，测试网络功能是否正常。

```
ping baidu.com
PING baidu.com (110.242.68.66): 56 data bytes
64 bytes from 110.242.68.66: seq=0 ttl=47 time=209.845 ms
64 bytes from 110.242.68.66: seq=1 ttl=47 time=203.668 ms
64 bytes from 110.242.68.66: seq=2 ttl=47 time=116.356 ms
64 bytes from 110.242.68.66: seq=3 ttl=47 time=126.607 ms
64 bytes from 110.242.68.66: seq=4 ttl=47 time=199.377 ms
```

若网络功能不正常，需在"/etc/resolv.conf"文件添加内容"nameserver 114.114.114.114"。可以用手机连接上述热点进行上网测试。

3.4. 音频接口

请准备一个带麦克风的耳机，连接 HP OUT/MIC IN 接口。

播放

```
#aplay /etc/test.wav
```

录音

将 mic_in_test.sh 脚本程序拷贝至评估板文件系统，进入评估板文件系统，执行如下命令，使用耳机的麦克风进行录音。按"Ctrl + C"终止录音后，会将录音文件保存为"test.wav"。

```
tinymix set "ADC1 Input FMINL Switch" 0
tinymix set "ADC1 Input MIC1 Boost Switch" 1
tinymix set "ADC1 Input LINEINL Switch" 0
tinymix set "ADC2 Input FMINR Switch" 0
tinymix set "ADC2 Input MIC2 Boost Switch" 0
tinymix set "ADC2 Input LINEINR Switch" 0
tinymix set "ADC3 Input MIC3 Boost Switch" 0
tinymix set "HpSpeaker Switch" 1
tinymix set "LINEOUT Switch" 0
tinymix set "MIC1 Input Select" 0
tinymix set "MIC2 Input Select" 0
tinymix set "MIC3 Input Select" 0
tinymix set "Headphone Switch" 1
tinymix set "Headphone volume" 7
arecord -Dhw:audiocodec -f S24_LE -r    16000    -vvv    test.wav
```

执行如下命令，耳机将正常播放"test.wav"文件的录音， 并且不存在杂音、失真现象，按"Ctrl + C"停止播放。

```
  channels      : 2
  rate          : 22050
  exact rate    : 22050 (22050/1)
  msbits        : 16
  buffer_size   : 11024
  period_size   : 2756
  period_time   : 124988
  tstamp_mode   : NONE
  tstamp_type   : MONOTONIC
  period_step   : 1
  avail_min     : 2756
  period_event  : 0
  start_threshold  : 11024
  stop_threshold   : 11024
  silence_threshold: 0
  silence_size  : 0
  boundary      : 1444937728
  appl_ptr      : 0
  hw_ptr        : 0
###############          +                                | 46%^C
Aborted by signal Interrupt...
#######################  +                                | 46%
```

4. 参考资料

- Linux kernel 开源社区：

  https://www.kernel.org/

- 全志 SDK 模块开发指南

5. 修订说明

修订说明表

版本	修改内容	修改时间	编制	校对	审批
V1.0	初稿	24-11-12	HSL	WYQ	WFX
V1.1	修改 2.2.3 章节内容	24-11-21	HSL	WYQ	WFX
V1.2	修改 2.2.3 章节内容	24-12-17	HSL	WYQ	WFX

6. 关于我们

销售热线：4000-330-990

技术支持：support@cdebyte.com 官方网站：https://www.ebyte.com

公司地址：四川省成都市高新西区西区大道 199 号 B5 栋

成都亿佰特电子科技有限公司
Chengdu Ebyte Electronic Technology Co.,Ltd.