



浙江工业大学

实验报告

课程：嵌入式系统 A

第七次实验

姓 名 凌智城

学 号 201806061211

专业班级 通信工程 1803 班

老 师 黄国兴

学 院 信息工程学院

提交日期 2021 年 6 月 16 日

实验 11：视频采集播放实验

1 实验目的

1. 了解视频传输及回放的原理；
2. 了解共享内存在视频存储中的作用；
3. 了解视频解码模块的实现方式。

2 实验内容

1. 掌握视频存储模块的实现方式；
2. 掌握视频回放显示的实现方法；
3. 编写视频解码回放的启动脚本。

3 实验步骤

步骤 1: 将试验箱主板和底板上的部件按下图方式连接；接上 CCD 摄像头，将 SD 卡放入卡槽：

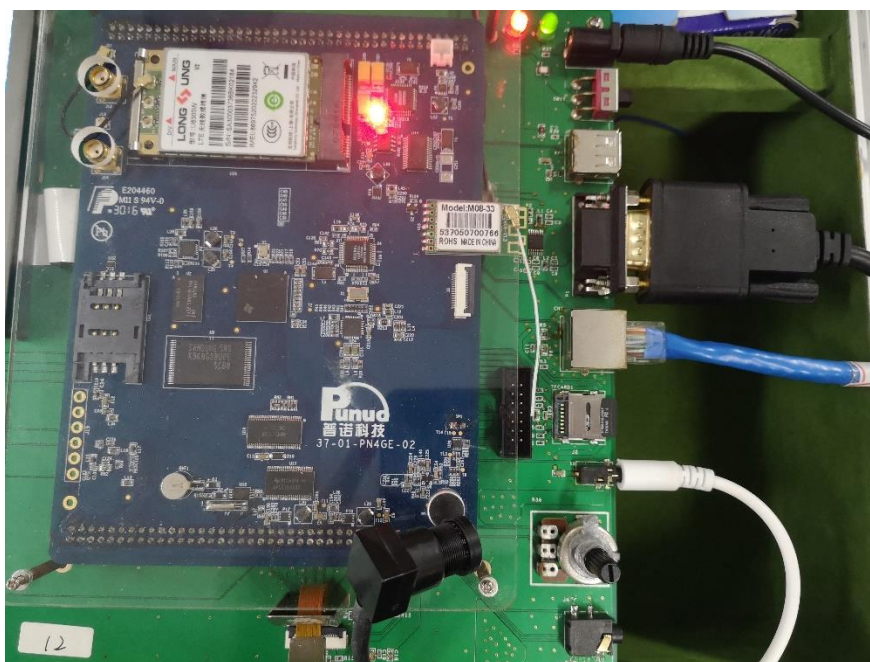


图 7-1 设备连接图

步骤 2: 查看设备端口号，选择设备管理器的端口查看



图 7-2 设备端口号查看方式

```
[root@zjut video]# df
Filesystem          1k-blocks      Used Available Use% Mounted on
tmpfs                43692          784    42908    2% /tmp
tmpfs                10240           32    10208    0% /dev
tmpfs                43692           0    43692    0% /dev/shm
/dev/mmcblk0p1      7628800      17504   7611296    0% /mnt/mmc
[root@zjut video]#
```

图 7-3 df 查看 SD 卡是否挂载成功

步骤 3: 在 putty 软件上设置端口环境：打开 putty，点击 Serial，在 Speed（波特率）栏设置为 115200，在 Serial line 栏输入上一步检测到的端口号（COM5），保存所设置的环境后进行挂在，此次挂在参数中有所更改（mem=90M）

```
setenv bootargs 'mem=90M console=ttyS0,115200n8 root=/dev/nfs rw
nfsroot=192.168.1.152:/home/shiyan/share/filesys_test
ip=192.168.1.52:192.168.1.152:192.168.1.1:255.255.255.0::eth0:off eth=00:40:01:C1:56:02
video=davinci_fb:vid0=OFF:vid1=OFF:osd0=640x480x16,600K:osd1=0x0x0,0K dm365_imp.oper_mode=0
davinci_capture.device_type=1 davinci_enc_mgr.ch0_output=LCD'
```

步骤 4: 在 COM 端口输入命令 “cd /mnt/mmc/video/”，按 ls 查看里面视频

```
[root@zjut ~]# cd /mnt/mmc/video/
[root@zjut video]# ls
10086.264          2017-06-17-10-56-08.264  2018-10-13-08-16-10.264
10086264          2017-06-17-10-58-47.264  2018-10-14-08-23-42.264
2017-01-26-06-34-05.264  2017-06-17-10-59-07.264  decode_D1.log
2017-01-26-07-10-38.264  2018-06-29-10-19-29.264
2017-01-26-07-11-33.264  2018-10-11-14-01-03.264
[root@zjut video]#
```

图 7-4 原来再 video 中有的视频文件

步骤 5: 开启实验箱的录像功能，点击试验箱的左下角按键第二排最后一个（2，4）按键，进入摄像界面，（点击试验箱的左下角按键第四排最后一个（4，

4) 按键, 可以切换 SSD 摄像头), 出现画面后点击屏幕上的开始录像, 录下来的视频数据会保存到实验箱的本地 SD 卡内

```
[root@zjut video]# ls
10086.264          2017-06-17-10-56-08.264  2018-10-13-08-16-10.264
10086264          2017-06-17-10-58-47.264  2018-10-14-08-23-42.264
2017-01-26-06-34-05.264  2017-06-17-10-59-07.264  2021-07-16-13-58-52.264
2017-01-26-07-10-38.264  2018-06-29-10-19-29.264  decode_D1.log
2017-01-26-07-11-33.264  2018-10-11-14-01-03.264
[root@zjut video]#
```

图 7-5 录制视频后产生了视频文件

视频文件 2021-07-16-13-58-52.264 为录制视频, 视频文件存储成功。

步骤 6: 编写视频解码回放脚本: 在服务器的 code 文件夹下有编写好的运行程序(dec_zh.sh, decode_zhfinal), 将其复制到实验箱文件系统 filesys_test 的 /opt/dm365 的目录下, 进入/filesys_test/opt/dm365, 输入指令

虚拟机: “cp /home/shiyan/Desktop/shiyan/code/video/* /home/shiyan/share/filesys_test/opt/dm365 “

“cd /home/shiyan/share/filesys_test/opt/dm365”

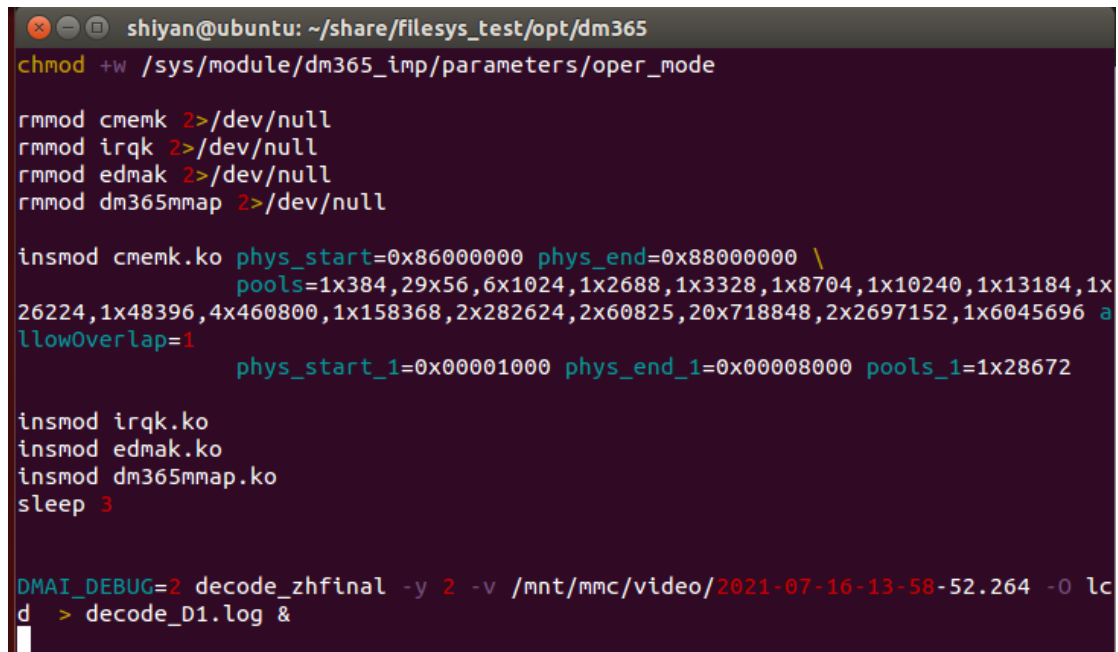
服务器: “cp /home/shiyan/2021/code/video/* /home/stX/filesys_test/opt/dm365”

```
shiyan@ubuntu: ~/share/filesys_test/opt/dm365
shiyan@ubuntu:~$ cp /home/shiyan/Desktop/shiyan/code/video/* /home/shiyan/share/filesys_test/opt/dm365
shiyan@ubuntu:~$ ls
Desktop  Downloads  kernel-for-mceb  Music  Public  Templates
Documents  examples.desktop  montavista  Pictures  share  Videos
shiyan@ubuntu:~$ cd ..
shiyan@ubuntu:/home$ ls
shiyan
shiyan@ubuntu:/home$ cd shiyan
shiyan@ubuntu:~$ cd share
shiyan@ubuntu:~/share$ cd filesys_test
shiyan@ubuntu:~/share/filesys_test$ cd opt
shiyan@ubuntu:~/share/filesys_test/opt$ cd dm365
shiyan@ubuntu:~/share/filesys_test/opt/dm365$ ls
3g_guard.sh      czzq          gpiotest      lcdtest        script
4g_mceb.sh      daemon        gps_app       led            sip_app
adctest         data          gpscfg.xml    led_on.bin     startup_mceb.sh
agc_check.sh    decode_zhfinal  guard_wcdma.sh  led_on_elf     task_db_1.sh
agc.sh          dec_zh.sh      hello         lm.sh          task_db_2.sh
amixer          dev1.pcap     helloworld    longPressKey   temp
a.out           dev_app_cl    i2c_test      ls             tvp2ov.sh
arecord         dev_app_pn    i2c_test_5151_1  myThread       uart57600
blend           dev.pcap      i2c_test_8bit   ov2tvp5151.sh  wlw
call            dm365mmap.ko  i2c_test_at24   play           wlwov
check_u6100     edmak.ko      image         pnrtc          wlw.tar.gz
check_u9600     encode        io            pollcsq
clear.sh        encode.log    iptables.sh    r_agc.sh
cmemk.ko        encode_mceb   irqk.ko        recv
Config.dat      getip.sh      lcd_evm        rtc_test
shiyan@ubuntu:~/share/filesys_test/opt/dm365$
```

图 7-6 将 decode_zhfinal 和 dec_zh.sh 复制到挂载文件系统的文件夹下

“cd /home/stX/filesys_test/opt/dm365”

查看 dec_zh.sh,运行: “vim dec_zh.sh”



```
shiyen@ubuntu: ~/share/filesys_test/opt/dm365
chmod +w /sys/module/dm365_imp/parameters/oper_mode

rmmod cmemk 2>/dev/null
rmmod irqk 2>/dev/null
rmmod edmak 2>/dev/null
rmmod dm365mmap 2>/dev/null

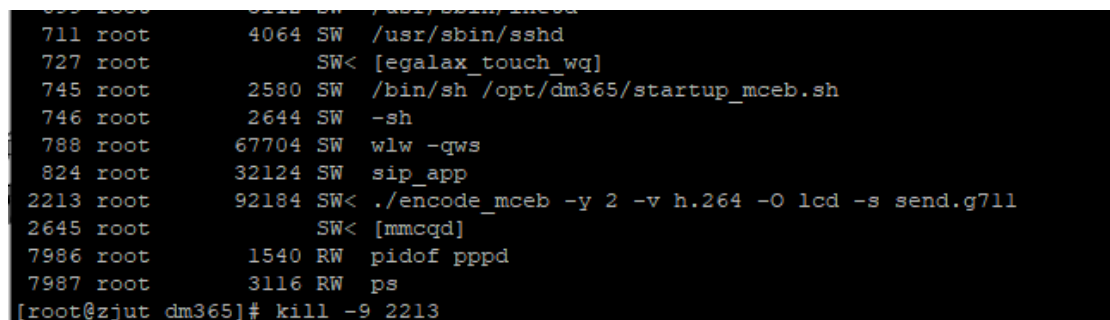
insmod cmemk.ko phys_start=0x86000000 phys_end=0x88000000 \
pools=1x384,29x56,6x1024,1x2688,1x3328,1x8704,1x10240,1x13184,1x
26224,1x48396,4x460800,1x158368,2x282624,2x60825,20x718848,2x2697152,1x6045696 a
llowOverlap=1
phys_start_1=0x00001000 phys_end_1=0x00008000 pools_1=1x28672

insmod irqk.ko
insmod edmak.ko
insmod dm365mmap.ko
sleep 3

DMAI_DEBUG=2 decode_zhfinal -y 2 -v /mnt/mmc/video/2021-07-16-13-58-52.264 -O lcd
d > decode_D1.log &
```

图7-7 更改脚本

以 root 身份进入板子文件系统的 /opt/dm365 目录, 输入命令 “cd /opt/dm365”, 然后输入命令: “ps“ 查看正在运行的进程, 如果显示有 encode 进程, 则需要 kill 掉: 输入命令: “kill -9 2213”



```
711 root      4064 SW   /usr/sbin/sshd
727 root      SW<  [egalax_touch_wq]
745 root      2580 SW   /bin/sh /opt/dm365/startup_mceb.sh
746 root      2644 SW   -sh
788 root      67704 SW  wlw -qws
824 root      32124 SW  sip_app
2213 root      92184 SW< ./encode_mceb -y 2 -v h.264 -O lcd -s send.g711
2645 root      SW<  [mmcqdl]
7986 root      1540 RW   pidof pppd
7987 root      3116 RW   ps
[root@zjut dm365]# kill -9 2213
```

图7-8 kill encode 进程

步骤 7: 点击第二排第四个 (2, 4) 按钮, 在板子 /opt/dm365 目录下执行解码回放脚本, 输入 “dec_zh.sh” 然后回车: (若出现 Permission denied, 运行 chmod 777 dec_zh.sh 即可) 当脚本执行的时候, 即开始了解码进程, 当按下板子上第二排最后一个按钮时候, 就可以看见解码的视频。视频左上角显示的 2021.07.16 13:59:04 是解码回放的视频的时间, 与之前解码脚本 dec_zh.sh 中写入的 /mnt/mmc/video/ 路径下的视频文件一一对应, 表示视频解码回放成功。


```

[root@zjut dm365]# dec_zh.sh
[ 1320.270000] irqk unregistered
[ 1320.480000] CMEMK module: built on Apr  7 2014 at 10:55:46
[ 1320.490000]   Reference Linux version 2.6.18
[ 1320.490000]   File /home/plc/dvSDK/linuxutils_2_24_02/packages/ti/sdo/linuxutils/cmem/src/module/cmemk.c
[ 1320.510000] ioremap_nocache(0x86000000, 33554432)=0xca000000
[ 1320.510000] allocated heap buffer 0xca000000 of size 0x480000
[ 1320.530000] cmem initialized 17 pools between 0x86000000 and 0x88000000
[ 1320.680000] IRQK module: built on Apr  7 2014 at 11:01:18
[ 1320.680000]   Reference Linux version 2.6.18
[ 1320.690000]   File /home/plc/dvSDK/linuxutils_2_24_02/packages/ti/sdo/linuxutils/irq/src/module/irqk.c
[ 1320.700000] irqk initialized
[ 1320.770000] EDMAK module: built on Apr  7 2014 at 10:58:36
[ 1320.770000]   Reference Linux version 2.6.18
[ 1320.780000]   File /home/plc/dvSDK/linuxutils_2_24_02/packages/ti/sdo/linuxutils/edma/src/module/edmak.c
[root@zjut dm365]# [ 1324.510000] DavinciDisplay DavinciDisplay.1: Before finishing with S_FMT:
[ 1324.510000] layer.pix_fmt.bytesperline = 640,
[ 1324.510000] layer.pix_fmt.width = 640,
[ 1324.510000] layer.pix_fmt.height = 480,
[ 1324.510000] layer.pix_fmt.sizeimage = 460800
[ 1324.540000] DavinciDisplay DavinciDisplay.1: pixfmt->width = 640,
[ 1324.540000] layer->layer_info.config.line_length= 640
[ 1338.680000] Alignment trap: decode_zhfinal (9076) PC=0x000153b8 Instr=0xe5933024 Address=0x001f3b23 FSR 0x001

```

图7-9 执行脚本解码视频



图7-10 回放采集到的视频

4 心得与体会

实践性比较强，综合了前几次实验的操作，通过这次视频采集播放实验，基本了解了视频采集的原理和摄像头采集的方式，实验设备的问题导致 SD 卡和实验箱经常出现问题，需要反复重启等操作。视频采集部分的代码量非常大，需要课前做足充分的准备并且在课后仔细研究视频采集部分的代码。