一 烧写事项

1. 在开发板不存在uboot的时候，我们利用Hitool工具来烧写uboot，kernel，文件系统；在开发板已存在uboot的时候我们可以采用tftp服务来烧些升级uboot，kernel，文件系统。
2. 在官方给的SDK包中已有制作好的根文件系统，可直接将其镜像文件烧写到开发板上。如果要添加自己的应用程序，将可执行文件，配置文件，库文件等拷贝到相应的目录即可。
3. 对于镜像文件的制作，可以直接到osdrv下直接使用make命令编译，也可单独编译。例如到osdrv/pub/bin/pc目录下去用下面的命令来制作yaffs2格式的系统镜像（Hi3559AV100支持yaffs2格式的文件系统）。

jffs2格式镜像

```
spi flash使用jffs2格式的镜像，制作jffs2镜像时，需要用到spi flash的块大小。这些信息会在uboot启动时会打印出来。建议使用时先直接运行mkfs.jffs2工具，根据打印信息填写相关参数。下面以块大小为64KB为例：
osdrv/pub/bin/pc/mkfs.jffs2 -d osdrv/pub/rootfs_glibc_xxx -l -e 0x40000 -o osdrv/pub/rootfs_hi3559av100_256k.jffs2
```

yaffs格式镜像

```
nand flash使用yaffs2格式的镜像，制作yaffs2镜像时，需要用到nand flash的pagesize和ecc。这些信息会在uboot启动时会打印出来。建议使用时先直接运行mkyaffs2image工具，根据打印信息填写相关参数。下面以2KB pagesize、4bit ecc为例：
osdrv/pub/bin/pc/mkyaffs2image100 osdrv/pub/rootfs_glibc_xxx osdrv/pub/rootfs_hi3559av100_2k_4bit.yaffs2 1 2
```

这里需要flash的pagesize 和 ecc信息，这些信息在uboot启动时会自动打印出来：

```
System startup
Uncompress Ok!
U-Boot 2016.11 (Nov 28 2019 - 10:19:42 +0800)hi3559av100
Relocation Offset is: 176ed000
Relocating to 5feed000, new gd at 5fe4ce00, sp at 5fe4cdf0
SPI Nor:  Boot Media isn't SPI Nor
NAND:  Check Flash Memory Controller v100 ... Found
SPI Nand ID Table Version 2.7
SPI Nand(cs 0) ID: 0xc8 0xc2 Name:"5F2GQ4RB9IGR"
Block:128KB Page:2KB OOB:128B ECC:24bit/1K
```

所以制作文件系统时出入的命令如下：

```
./mkyaffs2image100 ../../rootfs_glibc_m
ulti-core_arm64 rootfs_hi3559av100_2k_24bit.yadds2 2k 24bit
或者
./mkyaffs2image100 ../../rootfs_glibc_m
ulti-core_arm64 rootfs_hi3559av100_2k_24bit.yadds2 1 4
```

ext4格式镜像

```
emmc/ufs使用ext4格式的镜像：以96MB镜像为例：
make_ext4fs -l 96M -s osdrv/pub/rootfs_hi3559av100_96M.ext4
osdrv/pub/rootfs_glibc_xxx
```

4. 另外我们可以直接利用NFS挂载文件来进行相应的开发。对于文件系统我们可以在挂载后直接由cp（复制命令）来添加相关的文件。

二 启动参数设置

参数设置跟烧写时输入命令有关，关于地址的信息要一致。

1.单linux系统（multi-core）

> **SPI Nand Flash烧写启动参数设置**
>
> ```
> setenv bootargs 'mem=512M console=ttyAMA0,115200 root=/dev/mtdblock2 rw
> rootfstype=yaffs2 mtdparts=hinand:1M(boot),9M(kernel),16M(rootfs)'
>
> 注：mtdparts=hinand表示为Nand Flash，mtdparts=hi_sfc表示为SPI Nor FLASH
>
> setenv bootcmd 'nand read <ddr_addr> 0x100000 0x900000;bootm <ddr_addr>'
>
> saveenv
> ```
>
> **SPI Nor Flash烧写启动参数设置**
>
> ```
> setenv bootargs 'mem=64M console=ttyAMA0,115200 root=/dev/mtdblock2
> rootfstype=jffs2 rw mtdparts=hi_sfc:1M(boot),4M(kernel),16M(rootfs)'
> saveenv
> setenv bootcmd 'sf probe 0;sf read <ddr_addr> 0x100000 0x400000;bootm
> <ddr_addr>'
> saveenv
> ```
>
> 注释：<OS内存分配><设备串口号，波特率><第几个分区块><文件系统格式><flash类型>
>
> 注意：在设置uboot，kernel，roofts的大小时应注意和烧写时的配置一致，这样才能成功添加环境变量进入操作系统

2.双系统（big-little）

> ```
> setenv bootargs 'mem=512M console=ttyAMA0,115200 clk_ignore_unused rw
> root=/dev/mtdblock2 rootfstype=yaffs2
> mtdparts=hinand:1M(boot),9M(kernel),32M(rootfs)';sa
> (1)不启动M7
> setenv bootcmd 'nand read 0x45000000 0x2A00000 0x1000000;go_a53up
> 0x45000000; nand read 0x52000000 0x100000 0x900000;bootm 0x52000000';sa
> (2)启动M7
> setenv bootcmd 'nand read 0x45000000 0x2A00000 0x1000000;go_a53up
> 0x45000000; config_m7; nand read 0x52000000 0x3a00000 0x10000; cp.b
> 0x52000000 0x19000000 0x100000; go_m7; nand read 0x5200000 0x100000
> 0x900000;bootm 0x52000000';sa
> ```

三 Ubuntu 16.04系统上NFS的安装与使用

可参考以下链接进行NFS的安装：
https://blog.csdn.net/boling_cavalry/article/details/79498346

1. 赋予读写权限，创建(非必须) /home/winston/Hi3559AV100 目录

```
chmod a+rw /home/winston/Hi3559AV100
```

注意： 该路径是用来挂载到开发版上的，因此把需要挂载的路径赋予读写权限即可

2. 安装nfs服务器(服务器被客户端访问)

```
sudo apt-get update
sudo apt install nfs-kernel-server
```

3. 赋予挂载权限，打开 `/etc/exports`，加入下列内容(内容需修改)

```
/home/winston/Hi3559AV100 *(rw,sync,no_subtree_check,no_root_squash) 或
/home/winston/Hi3559AV100 *(rw,sync，no_root_squash)
*表示任何IP都可以访问，rw是读写权限，sync是同步权限，no_subtree_check表示不用管父目录是否有权限，no_root_squash表示不用
```

4. 重启NFS服务

```
sudo /etc/init.d/nfs-kernel-server restart
```

四 用NFS挂在文件时网络通信设置

在Linux下的终端下输入下面的命令

```
//设置开发板以太网物理地址
//设置网关，用于服务器和开发板处于不同网段时的相互通信
//设置开发板ip和掩码，需要和服务器在同一网段上
ifconfig eth0 hw ether FA:5A:E1:B1:89:75

ifconfig eth0 192.168.0.103 netmask 255.255.255.0

route add default gw 192.168.0.1
```

后续将设置网络通信和挂载的流程写到了自启动文件中，会自动挂载，然后取消了路由器，通过网线直连的方式

通过网线直连需要设置以太网的ip地址

- 首先输入`ifconfig`查看`ip`信息

```
eno1      Link encap:Ethernet  HWaddr e4:54:e8:9e:50:1d
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7431 errors:2 dropped:0 overruns:0 frame:1
          TX packets:24068 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:729966 (729.9 KB)  TX bytes:32788851 (32.7 MB)
```

```
          Interrupt:16 Memory:9dc80000-9dca0000

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:4778 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4778 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:388986 (388.9 KB)  TX bytes:388986 (388.9 KB)

wlxe84e06600d75 Link encap:Ethernet  HWaddr e8:4e:06:60:0d:75
          inet addr:192.168.1.116  Bcast:192.168.1.255
Mask:255.255.255.0
          inet6 addr: fe80::735b:6c79:b484:6af8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:113884 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16285 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17147832 (17.1 MB)  TX bytes:3632219 (3.6 MB)
```

- 以上的eno1为有线网ip，添加ip地址

```
ifconfig eno1 192.168.0.103
```

- 通过`ifconfig`查看ip信息

```
eno1      Link encap:Ethernet  HWaddr e4:54:e8:9e:50:1d
          inet addr:192.168.0.103  Bcast:192.168.0.255
Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7431 errors:2 dropped:0 overruns:0 frame:1
          TX packets:24092 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:729966 (729.9 KB)  TX bytes:32792707 (32.7 MB)
          Interrupt:16 Memory:9dc80000-9dca0000

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:4807 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4807 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:391155 (391.1 KB)  TX bytes:391155 (391.1 KB)

wlxe84e06600d75 Link encap:Ethernet  HWaddr e8:4e:06:60:0d:75
          inet addr:192.168.1.116  Bcast:192.168.1.255
Mask:255.255.255.0
          inet6 addr: fe80::735b:6c79:b484:6af8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:115130 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16361 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17275326 (17.2 MB)  TX bytes:3654428 (3.6 MB)
```

```
27
```

五 串口工具使用

minicom 的使用

minicom安装：`sudo apt-get install minicom`

minicom使用：

- 在终端输入 `ls -l  /dev/ttyUSB*` ，数显如下信息

  ```
  crw-rw---- 1 root dialout 188, 0 12月  2 10:00 /dev/ttyUSB0
  ```

  可以看到只有一个USB0挂载，并且具有读写权限（没有权限需要先给权限）输入`lsusb`查看串口是否连接上

- 输入`sudo minicom -s`进入minicom的调试配置，选择`serial port setup`。
  - 配置第一个选项`serial device`，填写相应的设备，这里我们填写`/dev/ttyUSB0`
  - 配置波特率，选择`115200`
  - 将`Hardware Flow Control`关闭
  - 选择`save setuo as dfl`进行保存
  - 利用`ctrl + x`退出minicom，再进入，`reset`开发板，考试使用串口与开发板通信

六 挂载共享目录

```
mount -t nfs -o nolock -o tcp -o rsize=32768,wsize=32768
192.168.0.100:/home/winston/Hi3559AV100/SDK020/Test/V100R001C02SPC020/01.soft
ware/board/Hi3559AV100_SDK_V2.0.2.0/ /mnt

mount -t nfs -o nolock -o tcp -o rsize=32768,wsize=32768
192.168.0.100:/home/winston/Hi3559AV100/SDK010/V100R001C02SPC010/01.software/
board/Hi3559AV100_SDK_V2.0.1.0/ /mnt

mount -t nfs -o nolock -o tcp -o rsize=32768,wsize=32768
192.168.8.105:/home/caizhi/Hi/Hi3559AV100_SDK030_LVDS /mnt
```

挂载时出现以下问题

```
~ # mount -t nfs -o nolock
192.168.0.100:/home/winston/Hi3559AV100/SDK020/Test/V
100R001C02SPC020/01.software/board/Hi3559AV100_SDK_V2.0.2.0 /mnt
mount: mounting
192.168.0.100:/home/winston/Hi3559AV100/SDK020/Test/V100R001C02SPC020/01.so
ftware/board/Hi3559AV100_SDK_V2.0.2.0 on /mnt failed: Connection timed out

解决办法：可能是防火墙的问题，关闭防火墙即可
    进入root模式下输入ufw disable （关闭防火墙）
                    ufw enable  （开启防护墙）

/mnt/mpp/out/linux/multi-core/ko # ./load3559av100_multicore -i -sensor0
imx334
```

```
sys_config: loading out-of-tree module taints kernel.
sys_config: Unknown symbol __stack_chk_guard (err 0)
sys_config: Unknown symbol __stack_chk_fail (err 0)
sys_config: Unknown symbol __stack_chk_guard (err 0)
sys_config: Unknown symbol __stack_chk_fail (err 0)
insmod: can't insert 'sys_config.ko': unknown symbol in module, or unknown parar
hi_osal: Unknown symbol __stack_chk_guard (err 0)
hi_osal: Unknown symbol __stack_chk_fail (err 0)
hi_osal: Unknown symbol __stack_chk_guard (err 0)
hi_osal: Unknown symbol __stack_chk_fail (err 0)
insmod: can't insert 'hi_osal.ko': unknown symbol in module, or unknown parametr
******* Error: There's something wrong, please check! *****

# 出现这个问题的原因是工具链不匹配或者板子上的SDK版本和交叉编译使用的不是一个版本
重新烧写对应 SDK 版本的 u-boot,kernel,文件系统
```

七 运行MPP业务

```
加载驱动：cd ./mpp/out/linux/multi-core/ko

            ./load3559av100_multicore -i -sensor0 imx334 -i sensor1 imx334
```

> 如果出现以下问题
>
> ```
> /mnt/mpp/out/linux/multi-core/ko # ./load3559av100_multicore -i sensor0 imx334
> -sh: ./load3559av100_multicore: Permission denied
> ```
>
> 需要执行以下命令
>
> ```
> cd /home/winston/Hi3559AV100/V100R001C02SPC010/01.software/board/Hi3559AV100_SDK_V2.0.1.0/mpp/out/linux/multi-core/ko
>
> chmod a+x load3559av100_multicore
> ```
>
> 加载驱动出现如下信息：
>
> ```
> /mnt/mpp/out/linux/multi-core/ko # ./load3559av100_multicore -i sensor0 imx327
> sys_config: loading out-of-tree module taints kernel.
> Module himedia: init ok
> ```
>
> insmod出现loading out-of-tree module taints kernel
>
> > rcS无可执行权限，`chmod u+x /etc/init.d/rcS`

运行例程

```
    cd ./mpp/sample/vio
    ./sample_vio 0 0
```

> 运行例程出现蓝屏的可能原因
>
> 1. 分辨率和帧率未能正确设置，需要与显示器的参数一致，具体修改sample_vio.c中最初的分辨率设置
>
> ```
> HI_S32 SAMPLE_VIO_8K30_PARALLEL(VO_INTF_TYPE_E enVoIntfType)
> {
>     HI_S32                  s32Ret              = HI_SUCCESS;
>     VI_DEV                  ViDev0              = 0;
>     ... ...
>     VO_CHN                  VoChn               = 0;
>     VO_INTF_SYNC_E          g_enIntfSync        =
> VO_OUTPUT_1080P60;
>     HI_U32                  g_u32DisBufLen      = 3;
>     PIC_SIZE_E              enPicSize           = PIC_1080P;
>
> ```
>
> 2. 尝试在sample目录下重新编译，先用`make linuxclean`清楚相关中间文件，再用`make linux`编译

八 TFTP服务烧写方案

8.1 tftp服务安装与配置

1. 安装tftp-server

```
sudo apt-get install tftpd-hpa
sudo apt-get install tftp-hpa  （如果不需要客户端可以不安装）
sudo apt-get install netkit-inetd  //  tftp 是要 inetd 来控制的, ubuntu 或 debian 类的系统,默认是没有安装 inetd 的
```

2. 配置tftp服务器

```
sudo vim /etc/default/tftpd-hpa
```

将内容改为

```
# /etc/default/tftpd-hpa
TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/home/winston/tftpboot"    //创建的下载目录，需要设置执行权限
TFTP_ADDRESS="192.168.0.100:69"
TFTP_OPTIONS="-l -c -s"
```

3. 创建下载目录

```
sudo mkdir /home/winston/tftpboot
```

4. 修改下载目录权限

```
sudo chmod 777 -R /home/winston/tftpboot/
```

5. 重新启动tftp服务

```
sudo /etc/init.d/tftpd-hpa restart
```

6. 测试tftp服务

   在/tftpboot目录下新建一个test文本文件，输入一些内容，然后回到主目录，

```
tftp 192.168.0.100  （本地ip）
tftp > get test.txt
tftp > q
回到根目录下
cat test
```

   若显示test文本的内容则配置成功。

7. 未正确烧写文件，应该将防火墙先关闭，在`root`权限下输入`ufw disable`关闭防火墙

   用tftp服务器烧录内核和文件系统时"Retry cout exceeded; starting again "解决方法

8. 如果烧写的时候一直等待状态，尝试输入`sudo /etc/init.d/tftpd-hpa restart`重启tftp服务。

8.2 烧写流程（单Linux）

- 配置tftp服务器，将发布包image_glibc_multi-core_arm64 目录下的相关文件拷贝到 tftp 服务器目录下。
- 参数配置，上电后，敲任意键进入u-boot。设置以下参数;

```
setenv ipaddr 192.168.0.103                 //单板ip
setenv ethaddr FA:5A:E1:B1:89:75          //单板的MAC地址
setenv netmask 255.255.255.0
setenv gatewayip 192.168.0.1
setenv serverip 192.168.0.100             //tftp服务器的ip
saveenv
ping 192.168.0.100                        //确保网络畅通。
```

烧写multi-core版本镜像文件到SPI Nand Flash（256M）

内存分配如下：

U-boot	Kernel	rootfs
1M	9M	16M

- 拨码选择主CPU：通过拨码开关SW1.4设置选择主CPU（Hi3559AV100 版本 A53UP 端仅支持 Huawei LiteOS）

  0：从A53MP COore0启动

  1：从A53UP启动

- 烧写u-boot

```
mw.b 0x44000000 0xff 0x100000
tftp 0x44000000 u-boot-hi3559av100.bin
nand erase 0x0 0x100000
nand write 0x44000000 0x0 0x100000
```

- 烧写内核

```
mw.b 0x44000000 0xff 0x900000
tftp 0x44000000 uImage_hi3559av100_multi-core
nand erase 0x100000 0x900000
nand write 0x44000000 0x100000 0x900000
```

- 烧写文件系统

```
mw.b 0x44000000 0xff 0x1000000
tftp 0x44000000 rootfs_hi3559av100_2k_24bit.yaffs2
nand erase 0xA00000 0x1000000
nand write.yaffs 0x44000000 0xA00000 0xbf5980 (0xbf5980 为 rootfs 文件实际大小)
```

- `reset`后设置启动参数

```
setenv bootargs 'mem=512M console=ttyAMA0,115200 root=/dev/mtdblock2 rw rootfstype=yaffs2 mtdparts=hinand:1M(boot),9M(kernel),16M(rootfs)'

setenv bootcmd 'nand read 0x44000000 0x100000 0x900000;bootm 0x44000000'

saveenv
```

- 重启系统

```
reset
```

烧写镜像到SPI Nor Flash（32M）

SPI Nor Flash 烧写时必须先对内存初始化，利用`sf probe`可进行`SPI Nor Flash`的探测和初始化。

内存分配如下：

U-boot	Kernel	roofts
1M	9M	15M

- 拨码选择主CPU：通过拨码开关SW1.4设置选择主CPU（Hi3559AV100 版本 A53UP 端仅支持 Huawei LiteOS）

  0：从A53MP COore0启动

  1：从A53UP启动

- 烧写U-boot

  在内存运行起来之后在终端输入

```
mw.b 0x41000000 0xff 0x100000                /* 对内存初始化*/
tftp 0x41000000 u-boot-hi3559av100.bin       /*U-boot下载到内存*/
sf probe 0                                   /*探测并初始化SPI flash*/
sf erase 0x0 0x100000                        /*擦除 1M大小*/
sf write 0x41000000 0x0 0x100000             /*从内存写入SPI NOR Flash,0x0
表示偏移地址*/
```

- 烧写kernel

```
mw.b 0x41000000 0xff 0x900000                      /* 对内存初始化*/
tftp 0x41000000 uImage_hi3559av100_multi-core      /*uImage下载到内存*/
sf probe 0                                         /*探测并初始化SPI flash*/
sf erase 0x100000 0x900000                         /*擦除 9M大小*/
sf write 0x41000000 0x100000 0x900000    /*从内存写入SPI NOR Flash,
0x1000000偏移地址*/
```

- 烧写roofts

```
mw.b 0x41000000 0xff 0x1000000                     /* 对内存初始化*/
tftp 0x41000000 rootfs_hi3559av100_64k.jffs2       /*文件系统下载到内存*/
sf probe 0                                         /*探测并初始化SPI flash*/
sf erase 0xa00000 0x1000000                            /*擦除 16M大小*/
sf write 0x41000000 0xa00000 0x1000000       /*从内存写入SPI NOR Flash,
0xa000000偏移地址*/
```

- `reset`设置启动参数

```
//设置bootrags
setenv bootargs 'mem=128M console=ttyAMA0,115200 root=/dev/mtdblock2
rootfstype=jffs2 rw mtdparts=hi_sfc:1M(boot),9M(kernel),16M(rootfs)'
saveenv

//设置bootcmd
setenv bootcmd 'sf probe 0;sf read 0x41000000 0x100000 0x900000;bootm
0x41000000'
saveenv
```

- 重启系统

`reset`

九 SDK安装

9.1 软件包安装

- 配置默认使用 bash
  执行`sudo dpkg-reconfigure dash`选择 no
- 安装软件包，执行

```
sudo apt-get install make libc6:i386 lib32z1 lib32stdc++6 zlib1g-dev
libncurses5-dev ncurses-term libncursesw5-dev g++ u-boot-tools:i386
texinfo texlive gawk libssl-dev openssl bc
```

- 创建/etc/ld.so.preload 文件，并执行 `echo "" > /etc/ld.so.preload`，以解决 64bit linux server 上某些第三方库编译失败的问题。

9.2 安装SDK

```
rar x Hi3559AV100R001C02SPC010.part1.rar
```

注意: 将生成的问价夹名字修改为 V100R001C02SPC010，即去掉前面的Hi3559A

在 01.software/board/目录下连续解压，命令如下：

```
tar -zxf Hi3559AV100_SDK_V2.0.1.0.tgz
sudo ./sdk.unpack
解包时出现错误：
/sdk.unpack: 2: source: not found
./sdk.unpack: 4: ECHO: not found
./sdk.unpack: 6: WARN: not found
./sdk.unpack: 7: WARN: not found
./sdk.unpack: 8: ECHO: not found

这是由于脚本是基于bash的，而用的linux可能是dash或其他的，利用如下命令即可解压：
bash ./sdk.unpack（未执行默认配置bash这一步）
```

9.3 安装交叉编译器

进入到相关编译器目录，执行以下命令

```
//安装aarch64交叉编译器
tar -xzf aarch64-himix100-linux.tgz
sudo chmod +x aarch64-himix100-linux.install
source ./aarch64-himix100-linux.install


//安装arm-none-eabi
tar -xzf gcc-arm-none-eabi-4_9-2015q3.tgz
进入https://launchpad.net/gcc-arm-embedded/4.9/4.9-2015-q3-update
下载gcc-arm-none-eabi-4_9-2015q3-20150921-linux.tar.bz2
1）将下载的 gcc-arm-none-eabi-4_9-2015q3-20150921-linux.tar.bz2和发布包里的
gcc-arm-none-eabi-4_9-2015q3.install放到同一个目录下
2)执行如下命令：
   chmod +x gcc-arm-none-eabi-4_9-2015q3.install
   ./gcc-arm-none-eabi-4_9-2015q3.install
 执行 source /etc/profile 配置环境变量生效
```

9.4 编译SDK

(该步骤可能要在安装交叉编译工具之后才能顺利进行)

1. 进入`/osdrv/opensource/kernel/`目录，根据`readme_cn.txt`进行操作

   先进入 www.kernel.org ， 选择`https://www.kernel.org/pub/`选项 → `linux/` → `kernel/` → `v4.x/`

   按下`ctrl f`，搜索`linux-4.9.37.tar.gz`并下载

下载包放于`osdrv/opensource/kernel/`中，

2. 在linux服务器中进入`osdrv`的根目录,执行如下命令：

```
    cd opensource/kernel
    tar -zxf linux-4.9.37.tar.gz
    mv linux-4.9.37 linux-4.9.y
    cd linux-4.9.y
    patch -p1 < ../linux-4.9.37.patch
    cd ../
    tar -czf linux-4.9.y.tgz linux-4.9.y
    cd ../../
```

或者在osdrv目录执行下列命令：

```
make atf
```

如果报错，很可能是第一步相关配置中库没装好，重新检查一遍

继续，可在`osdrv/pub/`下生成u-boot、uimage和文件系统，选择24bit的文件系统：

```
make
```

编译mpp，进入到mpp/sample目录下，编译例程

```
//编译
make linux
//清楚编译产生的文件
make linuxclean
```

在mpp目录下编译出现问题：

```
cp ../cfg.mak.multicore ../cfg.mak
make[1]: Entering directory
'/home/winston/Hi3559AV100/SDK020/Test/V100R001C02SPC020/01.software/board/
Hi3559AV100_SDK_V2.0.2.0/mpp/sample/vio'
../Makefile.param:45: *** missing separator.  Stop.
make[1]: Leaving directory
'/home/winston/Hi3559AV100/SDK020/Test/V100R001C02SPC020/01.software/board/
Hi3559AV100_SDK_V2.0.2.0/mpp/sample/vio'
Makefile:28: recipe for target 'linuxclean' failed
make: *** [linuxclean] Error 2

解决办法：
1. 如果使用的是 单系统linux方案，需要将cfg.mak.multicore文件更名为cfg.mak，然后在
sample、tool目录下去编译

2. 如果使用的是 linux+liteOS双系统方案，则：
   a. 如果要编译A53UP LiteOS上运行的模块，如vi、vo、venc等，需要先将cfg.mak.single
文件更名为cfg.mak，再进行编译
   b. 如果要编译A73MP Linux上运行的模块，如SVP等，需要先将cfg.mak.biglittle文件更名
为cfg.mak，再进行编译

```

十 调试工具

10.1 telnet安装与使用

因为在运行例程的时候不能退出界面，退出的话会导致sensor断电，整个工程会退出，不能进行调试，所以需要配置telnet协议。**telnet**协议是TCP/IP协议族中的一员，是Internet远程登录服务的标准协议和主要方式。 它为用户提供了在本地计算机上完成远程主机工作的能力。 在终端使用者的电脑上使用**telnet**程序，用它连接到服务器。

- 在服务器上安装telnet

```
sudo apt-get update
sudo apt-get install xinetd telnetd
//重启机器或网络服务
sudo /etc/init.d/xinetd restart
```

- telnet使用

  首先在开发板上输入 `telnetd&`开启telnet服务，然后在服务器上输入`telnet IP`即可远程登录到开发板上

- 登录过程中的密码

  用户名输入**root**，密码为默认直接回车

相当于开辟两个窗口，当一个窗口运行程序的时候，另一个窗口可以通过命令来查看相关调试信息。调试信息采用了 Linux 下的 proc 文件系统，可实时反映当前系统的运行状态，所记录的信息可供问题定位及分析时使用。

10.2 文件目录

`/proc/umap`

10.3 文件清单

文件名称	描述
sys	记录当前 SYS 模块的使用情况。
vb	记录当前 VB 模块的 buffer 使用情况。
logmpp	记录当前各个模块的调试级别，内部调试用。
chnl	CHNL 模块状态。
vgs	视频缩放处理单元状态信息。
h265e	H.265 编码过程中，各通道的编码属性、状态以及历史信息统计。
h264e	H.264 编码过程中，各通道的编码属性、状态以及历史信息统计。
jpege	JPEG 编码过程中，各通道的编码属性、状态以及历史信息统计。
rc	编码通道的码流控制属性、状态以及历史信息统计。
rgn	视频叠加 OSD 的区域管理信息。
venc	视频编码器信息。
vdec	视频解码器信息。
vi	视频输入模块信息。
vo	视频输出模块信息。
vpss	视频预处理模块信息。
avs	视频任意角度拼接模块信息。
dis	视频防抖模块信息。
gdc	几何畸变矫正模块信息。
ai	音频输入通道信息。
ao	音频输出通道信息。
aenc	音频编码信息。
adec	音频解码信息。
acodec	Acodec 音量信息。

10.4 信息查看方法

- 在控制台上可以使用 cat 命令查看信息，例如`cat /proc/umap/venc`；也可以使用其他常用的文件操作命令，例如`cp /proc/umap/ ./ -rf`，将所`umap`下的`proc`文件拷贝到当前目录。
- 在应用程序中可以将上述文件当作普通只读文件进行读操作，例如`fopen、fread`等。
- 示例如下：

```
~ # cat /proc/umap/logmpp
-----LOG BUFFER STATE-------------------------------------------------
-
```

```
MaxLen ReadPos WritePos ButtPos
64(KB) 19689 19688 65026
-----CURRENT LOG LEVEL-----------------------------------------------
VB : 3
SYS : 3
REGION : 3
CHNL : 3
VDEC : 3
AVS : 3
VPSS : 3
VENC : 3
H264E : 3
JPEGE : 3
VFMW : 3
JPEGD : 3
VOU : 3
VI : 3
DIS : 3
RC : 3
AIO : 3
AI : 3
AO : 3
AENC : 3
ADEC : 3
VEDU : 3
ISP : 3
IVE : 3
VGS : 3
H265E : 3
GDC : 3
HDMI : 3
TDE : 3
```