# 总结

|  |
| --- |
| Fbmem：  switch设备:  switch\_dev\_register(struct switch\_dev \*sdev);  switch\_dev\_unregister(struct switch\_dev \*sdev);  //注册和注销switch设备  switch\_get\_state(struct switch\_dev \*sdev)  switch\_set\_state(struct switch\_dev \*sdev, int state);  //获取和改变switch设备的状态，通知android层 |

# Sys系统分析

## dev设备

|  |
| --- |
| root@gs703d:/ # ls -al /dev/graphics/\* //显存设备  crw-rw---- root graphics 29, 0 2018-03-09 09:28 fb0  crw-rw---- root graphics 29, 1 2018-03-09 09:28 fb1  root@gs703d:/ # ls -al /dev/hdmi  crw-r----- system system 10, 20 2018-03-09 09:28 hdmi  root@gs703d:/ # ls -al /dev/cvbs  crw-r----- system system 10, 22 2018-03-09 09:28 cvbs |

## Graphics: class

|  |
| --- |
| ==============================  /sys/class/graphics  挂载了两个fb设备  ==============================  # ls -al /sys/class/**graphics**  **lrwxrwxrwx root root fb0 -> ../../devices/platform/owlfb.0/graphics/fb0**  **lrwxrwxrwx root root fb1 -> ../../devices/platform/owlfb.3/graphics/fb1** |

## owlfb

### Device: platform device = owlfb.0: fb0

|  |
| --- |
| ==============================  /sys/devices/platform/owlfb.0  /sys/bus/platform/devices/owlfb.0  ==============================  # ls -al /sys/devices/platform/**owlfb.0**  **lrwxrwxrwx root root driver -> ../../../bus/platform/drivers/owlfb**  **lrwxrwxrwx root root subsystem -> ../../../bus/platform**  drwxr-xr-x root root graphics  -r--r--r-- root root modalias //platform:owlfb  drwxr-xr-x root root power  -rw-r--r-- root root uevent |
| # ls -al /sys/devices/platform/**owlfb.0**/graphics/**fb0**  **lrwxrwxrwx root root device -> ../../../owlfb.0**  **lrwxrwxrwx root root overlay -> ../../../owldss.0/overlay0**  **lrwxrwxrwx root root subsystem -> ../../../../../class/graphics**  -rw-r--r-- root root bits\_per\_pixel //32  -rw-r--r-- root root blank  -r--r--r-- root root blank\_hold //1  -rw-r--r-- root root console  -rw-r--r-- root root cursor  -r--r--r-- root root dev //29:0  -rw-rw---- system system enable\_3d //0  -rw-rw---- system system mem\_mode //user-0,1,2,3  -r--r--r-- root root name //owlfb  -r--r--r-- root root overlay\_hold //1  -rw-r--r-- root root pan //0,0  -rw-r--r-- root root rotate //0  -rw-r--r-- root root state //0  -r--r--r-- root root stride //7680  -rw-r--r-- root root uevent  /\*  MAJOR=29  MINOR=0  DEVNAME=fb0  \*/  -rw-r--r-- root root user\_hold //1  -rw-r--r-- root root virtual\_size //1920,1092  -rw-rw---- system system mode //S:1920x1080p-60  -rw-r--r-- root root modes  /\*  S:640x480p-60  S:720x576p-50  S:720x480p-59  S:720x576i-50  S:720x480p-60  S:1280x720p-60  S:1280x720p-50  S:1920x1080i-60  S:1920x1080i-50  S:1920x1080p-24  S:1920x1080p-25  S:1920x1080p-30  S:1920x1080p-60  S:1920x1080p-50  S:1280x720p-59  S:1920x1080i-59  S:1920x1080p-59  S:3840x2160p-30  S:3840x2160p-25  S:3840x2160p-24  S:4096x2160p-24  \*/ |

### Device: platform device = owlfb.3: fb1:

|  |
| --- |
| ==============================  /sys/devices/platform/**owlfb.3**  /sys/bus/platform/devices/owlfb.3  ==============================  # ls -al /sys/devices/platform/owlfb.3  **lrwxrwxrwx root root driver -> ../../../bus/platform/drivers/owlfb**  **lrwxrwxrwx root root subsystem -> ../../../bus/platform**  drwxr-xr-x root root graphics  -r--r--r-- root root modalias //platform:owlfb  drwxr-xr-x root root power  -rw-r--r-- root root uevent |
| # ls -al /sys/devices/platform/**owlfb.3**/graphics/**fb1**  **lrwxrwxrwx root root device -> ../../../owlfb.0**  **lrwxrwxrwx root root overlay -> ../../../owldss.0/overlay0**  **lrwxrwxrwx root root subsystem -> ../../../../../class/graphics**  -rw-r--r-- root root bits\_per\_pixel //32  -rw-r--r-- root root blank  -r--r--r-- root root blank\_hold //0 //fb0 : 1  -rw-r--r-- root root console  -rw-r--r-- root root cursor  -r--r--r-- root root dev //29:1 // fb0 : 29:0  -rw-rw---- system system enable\_3d //0  -rw-rw---- system system mem\_mode //fbmem // fb0 : user-0,1,2,3  -r--r--r-- root root name //owlfb  -r--r--r-- root root overlay\_hold //0 // fb0 : 1  -rw-r--r-- root root pan //0,0  -rw-r--r-- root root rotate //0  -rw-r--r-- root root state //0  -r--r--r-- root root stride //2880 // fb0 : 7680  -rw-r--r-- root root uevent  /\*  MAJOR=29  MINOR=1 // fb0 : 0  DEVNAME=fb1 // fb0 : fb0  \*/  -rw-r--r-- root root user\_hold //0 // fb0 : 1  -rw-r--r-- root root virtual\_size //720,2919 // fb0 : 1920,1092  -rw-rw---- system system mode //S:720x576i-50 // fb0 : S:1920x1080p-60  -rw-r--r-- root root modes  /\*  S:720x576i-50  S:720x480i-60  \*/ |

### Driver : platform driver = owlfb

|  |
| --- |
| ==============================  sys/bus/platform/drivers/owlfb  ==============================  **lrwxrwxrwx root root owlfb.0 -> ../../../../devices/platform/owlfb.0**  **lrwxrwxrwx root root owlfb.3 -> ../../../../devices/platform/owlfb.3**  --w------- root root bind  --w------- root root unbind  --w------- root root uevent |

## owldss

### Device: platform device = owldss.0

|  |
| --- |
| ==============================  /sys/devices/platform/owldss.0  /sys/bus/platform/devices/owldss.0  ==============================  **lrwxrwxrwx root root driver -> ../../../bus/platform/drivers/owldss**  **lrwxrwxrwx root root subsystem -> ../../../bus/platform**  -r--r--r-- root root displayers  /\*  hdmi-plugged  cvbs-unplugged  \*/  -rw-r--r-- root root dither\_en //1  -r--r--r-- root root modalias //platform:owldss  drwxr-xr-x root root overlay0  drwxr-xr-x root root overlay1  drwxr-xr-x root root overlay2  drwxr-xr-x root root overlay3  drwxr-xr-x root root power  -rw-r--r-- root root uevent |

#### Overlay0

|  |
| --- |
| ==============================  /sys/devices/platform/owldss.0/overlay0  ==============================  -rw-r--r-- root root arrange //full  -r--r--r-- root root available\_arrange //original full fit original-or-fit all-custom  -rw-r--r-- root root channel //0,primary  -rw-rw---- system system cur\_displayer //hdmi  -rw-rw---- system system effect //128,7,7  -r--r--r-- root root enable //1  -rw-r--r-- root root global\_alpha //255  -rw-r--r-- root root global\_alpha\_enable //0  -rw-r--r-- root root master\_dev //fb  -r--r--r-- root root name //vid  -r--r--r-- root root owner //fb0  drwxr-xr-x root root region0  -r--r--r-- root root rotate //0  -r--r--r-- root root size //1920,1080  ==============================  /sys/devices/platform/owldss.0/overlay0/region0  ==============================  -rw-r--r-- root root coordinate //0,0  -rw-r--r-- root root enable //1  -rw-r--r-- root root input\_size //1920,1080  -rw-r--r-- root root offset //0,0  -rw-r--r-- root root output\_size //1920,1080 |

#### Overlay1/2

|  |
| --- |
| ==============================  /sys/devices/platform/owldss.0/overlay0  ==============================  -rw-r--r-- root root arrange //full  -r--r--r-- root root available\_arrange //original full fit original-or-fit all-custom  -rw-r--r-- root root channel //0,overlay //overlay0 : 0,primary  -rw-rw---- system system cur\_displayer //hdmi  -rw-rw---- system system effect //128,7,7  -r--r--r-- root root enable //1  -rw-r--r-- root root global\_alpha //255  -rw-r--r-- root root global\_alpha\_enable //0  -rw-r--r-- root root master\_dev //null //overlay0 : fb  -r--r--r-- root root name //vid  -r--r--r-- root root owner //fb0  drwxr-xr-x root root region0  -r--r--r-- root root rotate //0  -r--r--r-- root root size //128,4 //overlay0 :1920,1080  ==============================  /sys/devices/platform/owldss.0/overlay0/region0  ==============================  -rw-r--r-- root root coordinate //0,0  -rw-r--r-- root root enable //0 //overlay0 :1  -rw-r--r-- root root input\_size //128,4 //overlay0 :1920,1080  -rw-r--r-- root root offset //0,0  -rw-r--r-- root root output\_size //1920,1080 |

#### Overlay3

|  |
| --- |
| ==============================  /sys/devices/platform/owldss.0/overlay3  ==============================  -rw-r--r-- root root arrange //full  -r--r--r-- root root available\_arrange //original full fit original-or-fit all-custom  -rw-r--r-- root root channel //0,overlay //overlay0 : 0,primary  -rw-rw---- system system cur\_displayer //hdmi  -rw-rw---- system system effect //128,7,7  -r--r--r-- root root enable //1  -rw-r--r-- root root global\_alpha //255  -rw-r--r-- root root global\_alpha\_enable //0  -rw-r--r-- root root master\_dev //overlay0/3 :fb //overlay1/2 :null  -r--r--r-- root root name //vid  -r--r--r-- root root owner //fb0  drwxr-xr-x root root region0  -r--r--r-- root root rotate //0  -r--r--r-- root root size //overlay0/3 :1920,1080 //overlay1/2 : 128,4  ==============================  /sys/devices/platform/owldss.0/overlay3/region0  ==============================  -rw-r--r-- root root coordinate //0,0  -rw-r--r-- root root enable //0 //overlay0 :1  -rw-r--r-- root root input\_size //overlay0/3 :1920,1080 //overlay1/2 :128,4  -rw-r--r-- root root offset //0,0  -rw-r--r-- root root output\_size 720,576 ///overlay0/1/2 :1920,1080 |

### Driver : platform driver = owldss

|  |
| --- |
| ==============================  /sys/bus/platform/drivers/owldss  ==============================  lrwxrwxrwx root root owldss.0 -> ../../../../devices/platform/owldss.0  --w------- root root bind  --w------- root root unbind  --w------- root root uevent |

## owl\_display: class

|  |
| --- |
| ==============================  /sys/class/owl\_display/  ==============================  lrwxrwxrwx root root HDMI -> ../../devices/virtual/owl\_display/HDMI  lrwxrwxrwx root root cvbs -> ../../devices/virtual/owl\_display/cvbs  lrwxrwxrwx root root owl\_dummy\_displayer -> ../../devices/virtual/owl\_display/owl\_dummy\_displayer |
| ==============================  /sys/devices/virtual/owl\_display/HDMI  ==============================  drwxr-xr-x root root power  -rw-r--r-- root root state //1  lrwxrwxrwx root root subsystem -> ../../../../class/owl\_display  -rw-r--r-- root root uevent |
| ==============================  /sys/devices/virtual/owl\_display/cvbs/  ==============================  drwxr-xr-x root root power  -rw-r--r-- root root state //0  lrwxrwxrwx root root subsystem -> ../../../../class/owl\_display  -rw-r--r-- root root uevent |
| ==============================  /sys/devices/virtual/owl\_display/owl\_dummy\_displayer/  ==============================  drwxr-xr-x root root power  -rw-r--r-- root root state //0  lrwxrwxrwx root root subsystem -> ../../../../class/owl\_display  -rw-r--r-- root root uevent |

## act\_cvbs

### Device: platform device = act\_cvbs

|  |
| --- |
| ==============================  /sys//bus/platform/devices/act\_cvbs  /sys/devices/platform/act\_cvbs  ==============================  lrwxrwxrwx root root driver -> ../../../bus/platform/drivers/act\_cvbs  lrwxrwxrwx root root subsystem -> ../../../bus/platform  -r--r--r-- root root modalias //platform:act\_cvbs  drwxr-xr-x root root power |

### Driver : platform driver = act\_cvbs

|  |
| --- |
| ==============================  /sys/bus/platform/drivers/act\_cvbs  ==============================  lrwxrwxrwx root root act\_cvbs -> ../../../../devices/platform/act\_cvbs  --w------- root root bind  --w------- root root unbind  --w------- root root uevent |

### Misc system device

|  |
| --- |
| ==============================  /sys/devices/virtual/misc/cvbs  /sys/class/misc/cvbs  ==============================  lrwxrwxrwx root root subsystem -> ../../../../class/misc  -rw-r--r-- root root cvbs\_colorbar  -rw-r--r-- root root cvbs\_dump  -rw-r--r-- root root cvbs\_mode //0  -r--r--r-- root root dev //10:22  drwxr-xr-x root root power  -rw-r--r-- root root uevent |

### switch system device

|  |
| --- |
| ==============================  /sys/devices/virtual/switch/cvbs  /sys/class/switch/cvbs  ==============================  -r--r--r-- root root name //cvbs  drwxr-xr-x root root power  -r--r--r-- root root state //0  lrwxrwxrwx root root subsystem -> ../../../../class/switch  -rw-r--r-- root root uevent |

## b02c0000.hdmi

### Device: platform device = b02c0000.hdmi

|  |
| --- |
| ==============================  /sys/bus/platform/devices/b02c0000.hdmi  /sys/devices/b02c0000.hdmi  ==============================  lrwxrwxrwx root root driver -> ../../bus/platform/drivers/gl5203\_hdmi  lrwxrwxrwx root root subsystem -> ../../bus/platform  -r--r--r-- root root modalias //platform:b02c0000.hdmi  drwxr-xr-x root root power  -rw-r--r-- root root uevent |

### Driver : platform driver = gl5203\_hdmi

|  |
| --- |
| ==============================  /sys/bus/platform/drivers/gl5203\_hdmi  ==============================  lrwxrwxrwx root root b02c0000.hdmi -> ../../../../devices/b02c0000.hdmi  --w------- root root bind  --w------- root root unbind  --w------- root root uevent |

### Misc system device

|  |
| --- |
| ==============================  /sys/devices/virtual/misc/hdmi  ==============================  -rw-rw-r-- root root config //16  -r--r--r-- root root dev //10:20  -rw-rw---- system system hdmi3d //0  -rw-rw---- root root phy\_config  drwxr-xr-x root root power  lrwxrwxrwx root root subsystem -> ../../../../class/misc  -rw-rw-r-- root root tv3d //0  -rw-r--r-- root root uevent |

### switch system device

|  |
| --- |
| ==============================  /sys/devices/virtual/switch/hdmi  ==============================  lrwxrwxrwx root root subsystem -> ../../../../class/switch  -rw-rw---- system system status  -r--r--r-- root root name //hdmi  drwxr-xr-x root root power  -r--r--r-- root root state //1  -rw-r--r-- root root uevent |

## atm7059-i2s\_hdmi\_spdif-audio

### Device: platform device = atm7059-i2s\_hdmi\_spdif-audio

|  |
| --- |
| ==============================  /sys/devices/platform/atm7059-i2s\_hdmi\_spdif-audio  ==============================  lrwxrwxrwx root root driver -> ../../../bus/platform/drivers/atm7059-i2s\_hdmi\_spdif-audio  lrwxrwxrwx root root subsystem -> ../../../bus/platform  -r--r--r-- root root modalias //platform:atm7059-i2s\_hdmi\_spdif-audio  drwxr-xr-x root root power  -rw-r--r-- root root uevent |

### Driver : platform driver = atm7059-i2s\_hdmi\_spdif-audio

|  |
| --- |
| ==============================  /sys/bus/platform/drivers/atm7059-i2s\_hdmi\_spdif-audio  ==============================  lrwxrwxrwx root root b02c0000.hdmi -> ../../../../devices/atm7059-i2s\_hdmi\_spdif-audio  --w------- root root bind  --w------- root root unbind  --w------- root root uevent |

## atm7059-pcm-audio

### Device: platform device = atm7059-pcm-audio

|  |
| --- |
| ==============================  /sys/devices/platform/atm7059-pcm-audio/  ==============================  lrwxrwxrwx root root driver -> ../../../bus/platform/drivers/atm7059-pcm-audio  lrwxrwxrwx root root subsystem -> ../../../bus/platform  -rw------- root root debug  -rw------- root root error  -r--r--r-- root root modalias //platform:atm7059-pcm-audio  drwxr-xr-x root root power  -rw-r--r-- root root uevent |

### Driver : platform driver = atm7059-pcm-audio

|  |
| --- |
| ==============================  /sys/bus/platform/drivers/atm7059-pcm-audio  ==============================  lrwxrwxrwx root root b02c0000.hdmi -> ../../../../devices/ atm7059-pcm-audio  --w------- root root bind  --w------- root root unbind  --w------- root root uevent |

# APP

## Framebuffer结构体

|  |
| --- |
| **struct fb\_fix\_screeninfo** {  char id[16]; /\*字符串形式的标识符 \*/  /\* fb缓冲内存的开始地址（物理地址），它一般是作为dma\_alloc\_writecombine的参数，该函数会将物理地址存放在该变量中\*/  unsigned long smem\_start; /\*fb缓冲的物理地址\*/  \_\_u32 smem\_len; /\* fb缓冲的长度,等于max\_xres \*max\_yres\*max\_bpp/8 \*/  \_\_u32 type; /\* 查看宏 FB\_TYPE\_ \*/  /\*  #define FB\_TYPE\_PACKED\_PIXELS 0 /\* Packed Pixels \*/  #define FB\_TYPE\_PLANES 1 /\* Non interleaved planes \*/  #define FB\_TYPE\_INTERLEAVED\_PLANES 2 /\* Interleaved planes \*/  #define FB\_TYPE\_TEXT 3 /\* Text/attributes \*/  #define FB\_TYPE\_VGA\_PLANES 4 /\* EGA/VGA planes \*/  #define FB\_TYPE\_FOURCC 5 /\* Type identified by a V4L2 FOURCC \*/  \*/  \_\_u32 type\_aux; /\* 分界，=0 \*/  \_\_u32 visual; /\* 屏幕的色彩模式，如FB\_VISUAL\_TRUECOLOR（真彩色）\*/  /\*  #define FB\_VISUAL\_MONO01 0 /\* Monochr. 1=Black 0=White \*/  #define FB\_VISUAL\_MONO10 1 /\* Monochr. 1=White 0=Black \*/  #define FB\_VISUAL\_TRUECOLOR 2 /\* True color \*/  #define FB\_VISUAL\_PSEUDOCOLOR 3 /\* Pseudo color (like atari) \*/  #define FB\_VISUAL\_DIRECTCOLOR 4 /\* Direct color \*/  #define FB\_VISUAL\_STATIC\_PSEUDOCOLOR 5 /\* Pseudo color readonly \*/  #define FB\_VISUAL\_FOURCC 6 /\* Visual identified by a V4L2 FOURCC \*/  \*/  \_\_u16 xpanstep; /\* 如果没有硬件 panning，=0 \*/  \_\_u16 ypanstep; /\* 如果没有硬件 panning，=0 \*/  \_\_u16 ywrapstep; /\* 如果没有硬件 panning，=0 \*/  \_\_u32 **line\_length**; /\* 一行的字节数 \*/  unsigned long mmio\_start; /\* 内存映射的I/O的开始位置 \*/  \_\_u32 mmio\_len; /\* 内存映射的I/O的长度 \*/  \_\_u32 accel; /\* = FB\_ACCEL\_NONE \*/  \_\_u16 capabilities; /\* see FB\_CAP\_\* \*/  \_\_u16 reserved[3];  }; |
| **struct fb\_var\_screeninfo** {  /\*\*\*\*\*\*\*\*visible resolution（实际屏幕）\*\*\*\*\*\*\*\*/  \_\_u32 xres; /\*定义屏幕一行有多少个像素点 \*/  \_\_u32 **yres**; /\*定义屏幕一列由多少个像素点 \*/    /\*\*\*\*\*\*\*\*virtual resolution（虚拟屏幕）\*\*\*\*\*\*\*\*/  \_\_u32 xres\_virtual; /\*虚拟屏幕一行有多少个像素点 \*/  \_\_u32 yres\_virtual; /\*虚拟屏幕一列由多少个像素点\*/  \_\_u32 xoffset; /\*虚拟到实际之间的行方向偏移 \*/  \_\_u32 yoffset; /\*虚拟到实际之间的列方向偏移\*/    \_\_u32 bits\_per\_pixel; /\*每像素位数(多少BPP)，单位为字节 \*/  \_\_u32 grayscale; /\*非0时指灰度\*/    /\*\*\*\*\*\*\*\*fb缓存的RGB位域\*\*\*\*\*\*\*\*\*\*/  struct fb\_bitfield red; /\* fb缓存的红色位域\*/  struct fb\_bitfield green; /\* fb缓存的绿色位域\*/  struct fb\_bitfield blue; /\* fb缓存的蓝色位域\*/  struct fb\_bitfield transp; /\*透明度 =0 \*/    \_\_u32 nonstd; /\*非标准像素格式时应该为非0值\*/    \_\_u32 activate; /\*参见宏FB\_ACTIVATE\_NOW \*/  /\*  #define FB\_ACTIVATE\_NOW 0 /\* set values immediately (or vbl)\*/  #define FB\_ACTIVATE\_NXTOPEN 1 /\* activate on next open \*/  #define FB\_ACTIVATE\_TEST 2 /\* don't set, round up impossible \*/  #define FB\_ACTIVATE\_MASK 15  \*/  \_\_u32 height; /\* 高度\*/  \_\_u32 width; /\* 宽度 \*/    \_\_u32 accel\_flags; /\*查看fb\_info.flags \*/    /\*\*\*\*\*\*\*\*\*\*\*\*这参数必须通过查看LCD数据手册得到\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\*像素时钟（皮秒），pixclock=1/Dclk=... \*/  \_\_u32 pixclock; /\* pixel clock in ps (pico seconds) \*/  /\* 行切换，从同步到绘图之间的延迟即HFPD(有效数据之后无效的像素的个数) ，对应于LCD数据手册的Hsyn的front-porch\*/  \_\_u32 left\_margin; /\* time from sync to picture \*/    /\*行切换，从绘图到同步之间的延迟即HBPD(Hsyn脉冲下降沿之后的无效像素的个数) ,对应于LCD数据手册的Hsyn的back-porch\*/  \_\_u32 right\_margin; /\* time from picture to sync \*/    /\*帧切换，从同步到绘图之间的延迟即VFPD(有效数据之后还要经历的无效行数(之后是下一帧数据)) ，对应于LCD数据手册的Vsyn的front-porch\*/  \_\_u32 upper\_margin; /\* time from sync to picture \*/  /\*帧切换，从绘图到同步之间的延迟即VBPD(Vsyn脉冲下降沿之后还要经历的无效行数) ，对应于LCD数据手册的Vsyn的back-porch \*/  \_\_u32 lower\_margin;  /\*水平同步的长度即HSPW(Hsyn信号的脉冲宽度)，对应于LCD数据手册的Hsyn的pulse Width \*/  \_\_u32 hsync\_len; /\* length of horizontal sync\*/  /\*垂直同步的长度即VSPW(Vsyn信号的脉冲宽度)，对应于LCD数据手册的Vsyn的pulse Width \*/  \_\_u32 vsync\_len; /\* length of vertical sync\*/    \_\_u32 sync; /\* 查看宏FB\_SYNC\_\*/  \_\_u32 vmode; /\* 查看宏FB\_VMODE\_ \*/  \_\_u32 rotate; /\*顺时钟旋转的角度 \*/  \_\_u32 colorspace; /\* colorspace for FOURCC-based modes \*/  \_\_u32 reserved[5];  }; |

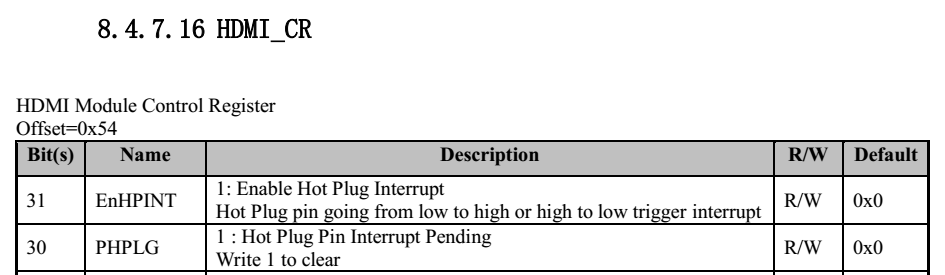
## 显示图像

|  |
| --- |
| char \*fb\_addr;  unsigned fb\_size;  1. 打开fb设备  int fbd=0;  fbd=open("/dev/fb0", O\_RDWR);  2. 获取fb的固定参数  struct fb\_fix\_screeninfo fb\_fix;  ioctl(fbd,FBIOGET\_FSCREENINFO,&fb\_fix);    3. 获取fb的可变参数  struct fb\_var\_screeninfo fb\_var;  ioctl(fbd,FBIOGET\_VSCREENINFO,&fb\_var);  4. 获取fb的缓存地址  fb\_size=fb\_var.yres\*fb\_fix.line\_length;  fb\_addr=(char \*)mmap(NULL,fb\_size,PROT\_READ|PROT\_WRITE,MAP\_SHARED, fbd,0);  5. 设置显示图像地址  short \*picture =(char \*)malloc(fb\_var.yres\*fb\_fix.line\_length);  memset(picture,0xFF,fb\_var.yres\*fb\_fix.line\_length);  6. 将显示图像的数据拷贝到fb的缓存地址显示  int width = fb\_var.xres;  int height = fb\_var.yres;  short \*src=(short \*) picture;  short \*dst=(short \*)fb\_addr;  int line=width<<1; /\*像素数乘以2即是字节数，因为颜色深度是2个字节(16bit)\*/  while(--height>=0)  {  memcpy(dst, src, line); /\*一行的数据赋值\*/  dst += width;  src += width;  } |

# HDMI

drivers\video\actions\display\hdmi\Atv5203\_hdmi.c

## HDMI的拔插中断



## struct hdmi\_sink\_info

|  |
| --- |
| struct hdmi\_sink\_info sink\_info = {  .v\_settings = {  .vid = VID1280x720P\_60\_16VS9, //会被DTS中的“resolution”改变  .hdmi\_mode = HDMI\_MODE\_HDMI,  .hdmi\_src = DE,  .pixel\_encoding = VIDEO\_PEXEL\_ENCODING\_RGB,  .color\_xvycc = YCC601\_YCC709,  .deep\_color = DEEP\_COLOR\_24\_BIT,  .\_3d = \_3D\_NOT,  },  .a\_settings = {  .audio\_channel = 1,  .audio\_fs = 2,  .audio60958 = 1,  },  .sink\_cap = {  .hdmi\_mode = HDMI\_MODE\_HDMI,  .sink\_phy\_addr = 0,  .sink\_3d\_cap = 0,  },  } |

## DTS: hdmi设备

|  |
| --- |
| **hdmi**:**hdmi**@**b02c0000**{  compatible="actions,gl5203-hdmi";  resolution="VID1920x1080P\_60\_16VS9";  }; |

## owl\_dss\_init2()

|  |  |  |
| --- | --- | --- |
| **owl**\_**dss**\_**init2**  1. 注册hdmi驱动  |--hdmi\_tx\_init();  |--platform\_driver\_register(&owl\_hdmi\_driver);   |  | | --- | | static struct platform\_driver owl\_hdmi\_driver = {  .driver = {  .name = ASOC\_HDMI\_NAME, /\*"gl5203\_hdmi"\*/  .owner = THIS\_MODULE,  .of\_match\_table = of\_match\_ptr(gl520x\_hdmi\_of\_match),  },  .probe = hdmi\_driver\_probe,  .remove = hdmi\_driver\_remove,  .suspend = hdmi\_driver\_suspend,  .resume = hdmi\_driver\_resume,  .shutdown = hdmi\_driver\_shutdown,  }; |   |--hdmi\_status\_store\_init(1);  2. 注册HDMI显示设备  |--tx\_info.hdmi\_dev\_for\_fb = owl\_hdmi\_display\_device\_register();  3. 第一次调用HDMI事件处理函数  |--hdmi\_do\_hdp(NULL);    4. 创建HDMI中断函数，处理HDMI硬件拔插事件  //#define OWL\_IRQ\_HDMI OWL\_IRQ(46) //32+46  |--request\_irq(OWL\_IRQ\_HDMI, hdmi\_irq\_handler, IRQF\_SHARED, "hdmi\_tx", (void \*)(&hdmi\_tx\_miscdev));   |  | | --- | | root@gs703d:/ # cat /proc/interrupts | grep hdmi  78: 0 GIC hdmi\_tx | |

## hdmi\_driver\_probe()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **hdmi**\_**driver**\_**probe**(struct platform\_device \*pdev)  1. 设置hdmi\_video\_settings的模式: VID1920x1080P\_60\_16VS9  |--struct hdmi\_sink\_info \*psink\_info = &sink\_info;  |--of\_get\_hdmi\_pars(pdev, psink\_info)  |--of\_property\_read\_string(of\_node, "resolution", &resolution)  |--vid = string\_to\_data\_fmt(resolution); //返回data\_fmt  |--psink\_info->v\_settings.vid = vid; //vid = VID1920x1080P\_60\_16VS9  2. 使能HDMI各模块的时钟  |--module\_clk\_enable(MOD\_ID\_TV24M);  |--module\_clk\_enable(MOD\_ID\_HDMI);  |--module\_clk\_enable(MOD\_ID\_HDMIA);  3. 创建HDMI的misc device : /sys/devices/virtual/misc/hdmi  |--misc\_register(&hdmi\_tx\_miscdev);  |--hdmi\_device\_attribute\_init();   |  | | --- | | struct miscdevice hdmi\_tx\_miscdev = {  .minor = 20,  .name = "hdmi",  .fops = &hdmi\_tx\_fops,  }; | | static const struct file\_operations hdmi\_tx\_fops = {  .owner = THIS\_MODULE,  .unlocked\_ioctl = hdmi\_ioctl,  .poll = hdmi\_poll,  }; |   4. 创建HDMI的switch device: /sys/devices/virtual/switch/hdmi  |--switch\_dev\_register(&hdev);   |  | | --- | | struct switch\_dev hdev = {  .name = "hdmi",  }; |   5. 创建HDMI的switch device: /sys/devices/virtual/switch/hdmi\_audio  |--switch\_dev\_register(&hdev\_audio);   |  | | --- | | struct switch\_dev hdev\_audio = {  .name = "hdmi\_audio",  }; |     6. 创建WORK线程  |--struct delayed\_work hdmi\_hdp\_work;  |--INIT\_DELAYED\_WORK(&hdmi\_hdp\_work, hdmi\_do\_hdp);  |--schedule\_delayed\_work(&hdmi\_hdp\_work, msecs\_to\_jiffies(0)); |

## owl\_hdmi\_display\_device\_register()： 注册fb显示设备

|  |
| --- |
| struct owl\_display\_device \***owl**\_**hdmi**\_**display**\_**device**\_**register**(void)  1. 创建并注册显示设备 : HDMI  // owl\_display\_device\_register(name,dev,devdata,display\_id,modes,num\_modes,ops)  |--return owl\_display\_device\_register("HDMI",  hdmi\_tx\_miscdev. //参见hdmi\_driver\_probe()  this\_device, NULL,  HDMI\_DISPLAYER, /\*display\_id\*/  hdmi\_display\_modes,  ARRAY\_SIZE(hdmi\_display\_modes),  &hdmi\_display\_ops); |

### HDMI : owl\_videomode

|  |
| --- |
| const struct owl\_videomode hdmi\_display\_modes[] = {  [0] = {  .valid = 1,  .mode = {  .name = "HDMI640x480P\_60\_4VS3",  .refresh = 60,  .xres = 640,  .yres = 480,  .pixclock = 39683, /\*pico second, 1.e-12s \*/  .left\_margin = 48,  .right\_margin = 16,  .upper\_margin = 33,  .lower\_margin = 2,  .hsync\_len = 96,  .vsync\_len = 2,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID640x480P\_60\_4VS3,  }  },  [1] = {  .valid = 1,  .mode = {  .name = "HDMI720x576P\_50\_4VS3",  .refresh = 50,  .xres = 720,  .yres = 576,  .pixclock = 37037, /\*pico second, 1.e-12s \*/  .left\_margin = 68,  .right\_margin = 12,  .upper\_margin = 39,  .lower\_margin = 5,  .hsync\_len = 64,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x576P\_50\_4VS3,  }  },  [2] = {  .valid = 1,  .mode = {  .name = "HDMI720x576P\_50\_16VS9",  .refresh = 50,  .xres = 720,  .yres = 576,  .pixclock = 37037, /\*pico second, 1.e-12s \*/  .left\_margin = 68,  .right\_margin = 12,  .upper\_margin = 39,  .lower\_margin = 5,  .hsync\_len = 64,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x576P\_50\_16VS9,  }  },  [3] = {  .valid = 1,  .mode = {  .name = "HDMI720x480P\_59P94\_4VS3",  .refresh = 59,  .xres = 720,  .yres = 480,  .pixclock = 37037, /\*pico second, 1.e-12s \*/  .left\_margin = 60,  .right\_margin = 16,  .upper\_margin = 30,  .lower\_margin = 9,  .hsync\_len = 62,  .vsync\_len = 6,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x480P\_59P94\_4VS3,  }  },  [4] = {  .valid = 1,  .mode = {  .name = "HDMIVID720x480P\_59P94\_16VS9",  .refresh = 59,  .xres = 720,  .yres = 480,  .pixclock = 37037, /\*pico second, 1.e-12s \*/  .left\_margin = 60,  .right\_margin = 16,  .upper\_margin = 30,  .lower\_margin = 9,  .hsync\_len = 62,  .vsync\_len = 6,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x480P\_59P94\_16VS9,  }  },  [5] = {  .valid = 1,  .mode = {  .name = "HDMI\_720x576I\_50\_4VS3",  .refresh = 50,  .xres = 720,  .yres = 576,  .pixclock = 37037, /\*pico second, 1.e-12s \*/  .left\_margin = 138,  .right\_margin = 24,  .upper\_margin = 20,  .lower\_margin = 4,  .hsync\_len = 126,  .vsync\_len = 6,  .sync = 0,  .vmode = FB\_VMODE\_INTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x576I\_50\_4VS3,  }  },  [6] = {  .valid = 1,  .mode = {  .name = "HDMI\_720x576I\_50\_16VS9",  .refresh = 50,  .xres = 720,  .yres = 576,  .pixclock = 37037, /\*pico second, 1.e-12s \*/  .left\_margin = 138,  .right\_margin = 24,  .upper\_margin = 20,  .lower\_margin = 4,  .hsync\_len = 126,  .vsync\_len = 6,  .sync = 0,  .vmode = FB\_VMODE\_INTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x576I\_50\_16VS9,  }  },  [7] = {  .valid = 1,  .mode = {  .name = "HDMI\_720x480P\_60\_4VS3",  .refresh = 60,  .xres = 720,  .yres = 480,  .pixclock = 37000, /\*pico second, 1.e-12s \*/  .left\_margin = 60,  .right\_margin = 16,  .upper\_margin = 30,  .lower\_margin = 6,  .hsync\_len = 62,  .vsync\_len = 9,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x480P\_60\_4VS3,  }  },  [8] = {  .valid = 1,  .mode = {  .name = "HDMI\_720x480P\_60\_16VS9",  .refresh = 60,  .xres = 720,  .yres = 480,  .pixclock = 37000, /\*pico second, 1.e-12s \*/  .left\_margin = 60,  .right\_margin = 16,  .upper\_margin = 30,  .lower\_margin = 6,  .hsync\_len = 62,  .vsync\_len = 9,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x480P\_60\_16VS9,  }  },  [9] = {  .valid = 1,  .mode = {  .name = "HDMI\_1280x720P\_60\_16VS9",  .refresh = 60,  .xres = 1280,  .yres = 720,  .pixclock = 13468, /\*pico second, 1.e-12s \*/  .left\_margin = 220,  .right\_margin = 110,  .upper\_margin = 20,  .lower\_margin = 5,  .hsync\_len = 40,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1280x720P\_60\_16VS9,  }  },  [10] = {  .valid = 1,  .mode = {  .name = "HDMI\_1280x720P\_50\_16VS9",  .refresh = 50,  .xres = 1280,  .yres = 720,  .pixclock = 13468, /\*pico second, 1.e-12s \*/  .left\_margin = 220,  .right\_margin = 440,  .upper\_margin = 20,  .lower\_margin = 5,  .hsync\_len = 40,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1280x720P\_50\_16VS9,  }  },  [11] = {  .valid = 1,  .mode = {  .name = "HDMI\_1920x1080I\_60\_16VS9",  .refresh = 60,  .xres = 1920,  .yres = 1080,  .pixclock = 13468, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 88,  .upper\_margin = 31,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 10,  .sync = 0,  .vmode = FB\_VMODE\_INTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080I\_60\_16VS9,  }  },  [12] = {  .valid = 1,  .mode = {  .name = "HMDI\_1920x1080I\_50\_16VS9",  .refresh = 50,  .xres = 1920,  .yres = 1080,  .pixclock = 13468, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 528,  .upper\_margin = 31,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 10,  .sync = 0,  .vmode = FB\_VMODE\_INTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080I\_50\_16VS9,  }  },  [13] = {  .valid = 1,  .mode = {  .name = "HDMI\_1920x1080P\_24\_16VS9",  .refresh = 24,  .xres = 1920,  .yres = 1080,  .pixclock = 13468, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 638,  .upper\_margin = 36,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080P\_24\_16VS9,  }  },  [14] = {  .valid = 1,  .mode = {  .name = "VID1920x1080P\_25\_16VS9",  .refresh = 25,  .xres = 1920,  .yres = 1080,  .pixclock = 13468, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 528,  .upper\_margin = 36,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080P\_25\_16VS9,  }  },  [15] = {  .valid = 1,  .mode = {  .name = "HDMI\_1920x1080P\_30\_16VS9",  .refresh = 30,  .xres = 1920,  .yres = 1080,  .pixclock = 13468, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 88,  .upper\_margin = 36,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080P\_30\_16VS9,  }  },  [16] = {  .valid = 1,  .mode = {  .name = "HDMI\_1920x1080P\_60\_16VS9",  .refresh = 60,  .xres = 1920,  .yres = 1080,  .pixclock = 6734, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 88,  .upper\_margin = 36,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080P\_60\_16VS9,  }  },  [17] = {  .valid = 1,  .mode = {  .name = "HDMI\_1920x1080P\_50\_16VS9",  .refresh = 50,  .xres = 1920,  .yres = 1080,  .pixclock = 6734, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 528,  .upper\_margin = 36,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080P\_50\_16VS9,  }  },  [18] = {  .valid = 1,  .mode = {  .name = "HDMI\_1280X720P\_59P94\_16VS9",  .refresh = 59,  .xres = 1280,  .yres = 720,  .pixclock = 13481, /\*pico second, 1.e-12s \*/  .left\_margin = 220,  .right\_margin = 110,  .upper\_margin = 20,  .lower\_margin = 5,  .hsync\_len = 40,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1280X720P\_59P94\_16VS9,  }  },  [19] = {  .valid = 1,  .mode = {  .name = "VID1920x1080I\_59P94\_16VS9",  .refresh = 59,  .xres = 1920,  .yres = 1080,  .pixclock = 13481, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 88,  .upper\_margin = 31,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 10,  .sync = 0,  .vmode = FB\_VMODE\_INTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080I\_59P94\_16VS9,  }  },  [20] = {  .valid = 1,  .mode = {  .name = "HDMI\_1920x1080P\_59P94\_16VS9",  .refresh = 59,  .xres = 1920,  .yres = 1080,  .pixclock = 6741, /\*pico second, 1.e-12s \*/  .left\_margin = 148,  .right\_margin = 88,  .upper\_margin = 36,  .lower\_margin = 4,  .hsync\_len = 44,  .vsync\_len = 5,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID1920x1080P\_59P94\_16VS9,  }  },  [21] = {  .valid = 1,  .mode = {  .name = "VID3840x2160p\_30\_16VS9",  .refresh = 30,  .xres = 3840,  .yres = 2160,  .pixclock = 4018, /\*pico second, 1.e-12s \*/  .left\_margin = 0,  .right\_margin = 0,  .upper\_margin = 0,  .lower\_margin = 0,  .hsync\_len = 0,  .vsync\_len = 0,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID3840x2160p\_30\_16VS9,  }  },  [22] = {  .valid = 1,  .mode = {  .name = "VID3840x2160p\_25\_16VS9",  .refresh = 25,  .xres = 3840,  .yres = 2160,  .pixclock = 4822, /\*pico second, 1.e-12s \*/  .left\_margin = 0,  .right\_margin = 0,  .upper\_margin = 0,  .lower\_margin = 0,  .hsync\_len = 0,  .vsync\_len = 0,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID3840x2160p\_25\_16VS9,  }  },  [23] = {  .valid = 1,  .mode = {  .name = "VID3840x2160p\_24\_16VS9",  .refresh = 24,  .xres = 3840,  .yres = 2160,  .pixclock = 5023, /\*pico second, 1.e-12s \*/  .left\_margin = 0,  .right\_margin = 0,  .upper\_margin = 0,  .lower\_margin = 0,  .hsync\_len = 0,  .vsync\_len = 0,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID3840x2160p\_24\_16VS9,  }  },  [24] = {  .valid = 1,  .mode = {  .name = "VID4096x2160p\_24\_16VS9",  .refresh = 24,  .xres = 4096,  .yres = 2160,  .pixclock = 5023, /\*pico second, 1.e-12s \*/  .left\_margin = 0,  .right\_margin = 0,  .upper\_margin = 0,  .lower\_margin = 0,  .hsync\_len = 0,  .vsync\_len = 0,  .sync = 0,  .vmode = FB\_VMODE\_NONINTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID4096x2160p\_24\_16VS9,  }  },  } |

### HDMI : display\_ops

|  |
| --- |
| struct display\_ops **hdmi**\_**display**\_**ops** = {  .update\_status = hdmi\_display\_update\_status,  .get\_status = hdmi\_display\_get\_status,  .update\_mode = hdmi\_display\_update\_mode,  .get\_devclk\_khz = hdmi\_display\_get\_devclk\_khz,  .set\_preline = hdmi\_display\_set\_preline,  }; |

## hdmi\_irq\_handler(): HDMI拔插事件中断函数

|  |
| --- |
| **hdmi**\_**irq**\_**handler**(s32 irq, void \*dev\_id)  |--s32 reg\_cr\_val = act\_readl(HDMI\_CR);  |--if ((reg\_cr\_val & HDMI\_CR\_ENHPINT/\*b31\*/) && (reg\_cr\_val & HDMI\_CR\_PHPLG/\*b30\*/))  |--schedule\_delayed\_work(&hdmi\_hdp\_work, msecs\_to\_jiffies(0)); |

## hdmi\_do\_hdp(): HDMI拔插事件处理函数

|  |
| --- |
| **hdmi**\_**do**\_**hdp**(struct work\_struct \*work)  1. 读HDMI模块的寄存器，获取HDMI的状态  |--state = hdmi\_get\_plug\_state();  |--if (plug\_detect\_num < 3)  |--plug\_detect\_num++;  |--schedule\_delayed\_work(&hdmi\_hdp\_work, msecs\_to\_jiffies(20));  |--return;  |--else  |--plug\_detect\_num=0;  2. 若HDMI插入  |--if ((hdmi\_get\_plug\_state() == true) &&  (tx\_info.hdmi\_dev\_for\_fb->is\_plugged == HDMI\_PLUGOUT)) //plugged in  |--atomic\_set(&hdmi\_plugging, 1);  2.1 更新HDMI的mode  |--owl\_display\_device\_store\_modes(tx\_info.hdmi\_dev\_for\_fb,  hdmi\_display\_modes,  ARRAY\_SIZE  (hdmi\_display\_modes));  |--tx\_info.hdmi\_dev\_for\_fb->is\_plugged = HDMI\_PLUGIN;  2.2 通知display模块的回调函数: owl\_display\_notifier\_callback()  //event = 0 (ASOC\_DISPLAY\_NOTIF\_PLUG), flag = 1  |--owl\_display\_notifier\_call\_chain(0, &fist\_hdmi\_in);  |--fist\_hdmi\_in = 0;  2.3 设置hdmi swtich的状态，通知android层  |--switch\_set\_state(&hdev, 1);  |--switch\_set\_state(&hdev\_audio, HDMI\_AUDIO\_ENABLE);  3. 若HDMI拔出  |--else if ((hdmi\_get\_plug\_state() == false) &&  (tx\_info.hdmi\_dev\_for\_fb->is\_plugged == HDMI\_PLUGIN)) //plugged out  |--atomic\_set(&hdmi\_plugging, 0);  |--owl\_display\_device\_store\_modes(tx\_info.hdmi\_dev\_for\_fb,  hdmi\_display\_modes,  ARRAY\_SIZE  (hdmi\_display\_modes));  |--tx\_info.hdmi\_dev\_for\_fb->is\_plugged = HDMI\_PLUGOUT;  3.1 通知display模块的回调函数: owl\_display\_notifier\_callback()  //event = 0 (ASOC\_DISPLAY\_NOTIF\_PLUG), flag = NULL  |--owl\_display\_notifier\_call\_chain(0, NULL);  3.2 设置hdmi swtich的状态，通知android层  |--switch\_set\_state(&hdev, 0);  |--switch\_set\_state(&hdev\_audio, HDMI\_AUDIO\_DISABLE); |

# Cvbs

|  |
| --- |
| # ls -al /dev/cvbs  crw-r----- system system 10, 22 2018-03-08 10:41 cvbs |
| # ls -al /sys/class/misc/cvbs/\*  -rw-r--r-- root root 4096 2018-03-08 10:41 cvbs\_colorbar  -rw-r--r-- root root 4096 2018-03-08 10:41 cvbs\_dump  -rw-r--r-- root root 4096 2018-03-08 10:41 cvbs\_mode  -r--r--r-- root root 4096 2018-03-08 10:41 dev  -rw-r--r-- root root 4096 2018-03-08 10:41 autosuspend\_delay\_ms  -rw-r--r-- root root 4096 2018-03-08 10:41 control  -r--r--r-- root root 4096 2018-03-08 10:41 runtime\_active\_time  -r--r--r-- root root 4096 2018-03-08 10:41 runtime\_status  -r--r--r-- root root 4096 2018-03-08 10:41 runtime\_suspended\_time  lrwxrwxrwx root root 2018-03-08 10:41 subsystem -> ../../../../class/misc  -rw-r--r-- root root 4096 2018-03-08 10:41 uevent |

## act\_cvbs\_init()

driver\video\actions\display\tvout\actions\_tvout.c

|  |  |  |
| --- | --- | --- |
| owl\_dss\_init2(void)  |--act\_cvbs\_init(void)  1. 注册cvbs设备  |--platform\_device\_register(&act\_cvbs\_device);  2. 注册cvbs驱动  |--platform\_driver\_register(&act\_cvbs\_driver);   |  | | --- | | static struct platform\_device act\_cvbs\_device = {  .name = "act\_cvbs",  .id = -1,  }; | | static struct platform\_driver act\_cvbs\_driver = {  .driver = {  .name = "act\_cvbs",  .owner = THIS\_MODULE,  },  .probe = act\_cvbs\_probe,  .remove = act\_cvbs\_remove,  }; | |

## act\_cvbs\_probe()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| act\_cvbs\_probe(struct platform\_device \*pdev)  1. 注册cvbs misc device及其属性  |--ret = misc\_register(&cvbs\_miscdev);  |--for (t = 0; t < ARRAY\_SIZE(owl\_cvbs\_attrs); t++)  |--r = device\_create\_file(cvbs\_miscdev.this\_device, &owl\_cvbs\_attrs[t]);   |  | | --- | | static struct miscdevice cvbs\_miscdev = {  .minor = 22,  .name = "cvbs",  .fops = &cvbs\_fops,  }; | | static struct file\_operations cvbs\_fops = {  .owner = THIS\_MODULE,  .open = cvbs\_open,  .unlocked\_ioctl = cvbs\_ioctl,  .poll = cvbs\_poll,  }; |   2. 获取cvbs的状态和制式: enable: PAL, NTSC  |--cvbs\_boot\_inited();  |--enable = act\_readl(TVOUT\_EN);  |--msr = act\_readl(CVBS\_MSR);  |--if(enable == 0) return -1; //not enable  |--else  |--if(msr == 0x14) return TV\_MODE\_PAL; //PAL  |--if(msr == 0x10) return TV\_MODE\_NTSC; //NTSC  |--else return 0;  4. 注册switch设备  |--cvbs\_info.sdev.name = "cvbs";  |--switch\_dev\_register(&cvbs\_info.sdev);   |  | | --- | | struct cvbs\_info cvbs\_info= {  .tv\_mode = TV\_MODE\_PAL,  .cvbs\_dev = NULL,  .cvbs\_enabled = ATOMIC\_INIT(0),  .current\_vid = 0,  }; |   5. 注册cvbs显示设备  // owl\_display\_device\_register(name,dev,devdata,display\_id,modes,num\_modes,ops)  |--cvbs\_info.cvbs\_dev = owl\_display\_device\_register("cvbs",  cvbs\_miscdev.this\_device,  NULL,  TV\_CVBS\_DISPLAYER, /\*display\_id\*/  cvbs\_display\_modes,  ARRAY\_SIZE(cvbs\_display\_modes),  &cvbs\_display\_ops);   |  | | --- | | display\_id :  enum disp\_dev\_id {  LCD\_DISPLAYER = 0x1,  HDMI\_DISPLAYER = 0x2,  TV\_CVBS\_DISPLAYER = 0x4,  TV\_YPbPr\_DISPLAYER = 0x8,  LCD1\_DISPLAYER = 0x10,  DSI\_DISPLAYER = 0x20,  }; |   |--cvbs\_info.cvbs\_dev->disp\_cur\_mode = &cvbs\_display\_modes[cvbs\_info.tv\_mode].mode;  |--cvbs\_info.cvbs\_dev->is\_plugged = CVBS\_PLUGOUT; |

### cvbs : owl\_videomode

|  |
| --- |
| static const struct owl\_videomode cvbs\_display\_modes[] = {  [0] = {  .valid = 1,  .mode = {  .name = "CVBS-PAL",  .refresh = 50,  .xres = 720,  .yres = 576,  .pixclock = 74074, /\*pico second, 1.e-12s \*/  .left\_margin = 48,  .right\_margin = 16,  .upper\_margin = 33,  .lower\_margin = 2,  .hsync\_len = 1,  .vsync\_len = 1,  .sync = 0,  .vmode = FB\_VMODE\_INTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x576I\_50\_4VS3,  }  },  [1] = {  .valid = 1,  .mode = {  .name = "CVBS-NTSC",  .refresh = 60,  .xres = 720,  .yres = 480,  .pixclock = 74074, /\*pico second, 1.e-12s \*/  .left\_margin = 68,  .right\_margin = 12,  .upper\_margin = 39,  .lower\_margin = 5,  .hsync\_len = 1,  .vsync\_len = 1,  .sync = 0,  .vmode = FB\_VMODE\_INTERLACED,  .flag = FB\_MODE\_IS\_STANDARD,  .vid = VID720x480I\_60\_4VS3,  }  },  } |

### cvbs : display\_ops

|  |
| --- |
| static struct display\_ops cvbs\_display\_ops = {  .update\_status = act\_cvbs\_update\_status,  .get\_status = act\_cvbs\_get\_status,  .update\_mode = act\_cvbs\_update\_mode,  .get\_devclk\_khz = act\_cvbs\_get\_devclk\_khz,  .set\_preline = act\_cvbs\_set\_preline,  }; |

# Dss: “owldss”

## owl\_dss\_init2

driver\video\actions\dss\main.c

driver\video\actions\dss\core.c

driver\video\actions\dss\dss.c

module\_init(owl\_dss\_init2);

|  |  |
| --- | --- |
| owl\_dss\_init2  1. 初始化cvbs  |--act\_cvbs\_init();  2. 初始化hdmi  |--hdmi\_tx\_init();  |--hdmi\_status\_store\_init(1);  3.注册owldss设备  |--static struct platform\_device \*owl\_dss\_devices;  //platform\_device\_register\_simple (name, id, res, num)  |--owl\_dss\_devices = platform\_device\_register\_simple("owldss", 0, NULL, 0);  |--device\_enable\_async\_suspend(&owl\_dss\_devices->dev);  4.注册owldss驱动  |--platform\_driver\_register(&owl\_dss\_driver);   |  | | --- | | struct platform\_driver owl\_dss\_driver = {  .probe = owl\_dss\_probe,  .remove = owl\_dss\_remove,  .shutdown = owl\_dss\_shutdown,  .suspend = owl\_dss\_suspend,  .resume = owl\_dss\_resume,  .driver = {  .name = "owldss",  .owner = THIS\_MODULE,  },  }; | |

## de\_init()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **de\_init**()  1. 添加DE的hw操作函数接口  |--de\_layer\_hw\_ops\_init();  |--for (i = 0; i < MAX\_OVERLAY\_NUM; i++)  |--de\_layer\_hw\_ops\_register(i, &video\_hw\_ops);  |--owl\_de.hw\_ops[layer\_id] = ops;   |  | | --- | | struct de\_layer\_hw\_ops video\_hw\_ops = {  .set\_region = de\_v\_set\_region,  //de\_layer\_apply(struct owl\_overlay \*ovl, bool force)  .flip\_region = de\_v\_flip\_region,  //de\_layer\_flip(struct owl\_overlay \*ovl)  }; |   2. 创建DE支持的4个overlay，添加到/sys/devices/platform/owldss.0/overlay[0123]中  |--for (i = 0; i < MAX\_OVERLAY\_NUM/\*4\*/; i++)  //dss\_overlay\_register(name,id,supported\_pixfmts,caps,num\_regions)  |--owl\_de.layers[i] = **dss\_overlay\_register**("vid", i, GL5201\_VIDEO\_PIXFMT,  ASOC\_DSS\_OVL\_CAP\_SCALE | ASOC\_DSS\_OVL\_CAP\_ROT\_180,  VIDEO\_REGION\_NUM/\*1\*/);  3. 创建dummy显示设备，注册到class/owl\_display/owl\_dummy\_displayer子系统中  |--for (i = 0; i < MAX\_CHANNEL\_NUM/\*2\*/; i++)  |--owl\_de.channels[i] = dss\_channel\_register(i, 0);  |--struct owl\_channel \*channel = kzalloc(sizeof(\*channel), GFP\_KERNEL);  |--channel->id = id; //0,1  |--**owl\_channel\_set\_device(channel, DUMMY\_DISPLAYER);**  |--**owl\_dss\_add\_channel(channel);**  4. DE控制器的初始化  |--global\_info\_init();  |--owl\_de.global\_info->info\_dirty = false;  |--de\_clk\_init();  |--de\_reset\_de();  |--de\_hw\_init();  5. 增加DE控制器的isr线程函数和irq中断函数  |--init\_de\_irq();  5.1 创建线程 : de-isr[012]  |--for (i = 0; i < DE\_ISR\_GROUP\_NUM/\*3\*/; i++)  |--delay\_work = &owl\_de.de\_work[i];  |--snprintf(wkq\_name, sizeof(wkq\_name), "de-isr%d", i);  |--delay\_work->workqueue = create\_singlethread\_workqueue(wkq\_name);  |--INIT\_WORK(&delay\_work->work, owl\_de\_work);  5.2 创建中断函数 : owl\_de, 33  |--request\_irq(OWL\_IRQ\_DE, de\_handle\_irq, IRQF\_DISABLED, "owl\_de", &owl\_de)   |  | | --- | | arch\arm\mach-owl\include\mach\irqs.h  #define OWL\_IRQ\_DE OWL\_IRQ(1) /\*33\*/ | | root@gs703d:/sys # ps | grep de-isr  root 67 2 0 0 c004aa14 00000000 S de-isr0  root 68 2 0 0 c004aa14 00000000 S de-isr1  root 69 2 0 0 c004aa14 00000000 S de-isr2 | | root@gs703d:/sys # cat /proc/interrupts | grep owl\_de  33: 151 GIC owl\_de | |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

|  |
| --- |
|  |

# End