

doi: 10.3969/j.issn.1006-1576.2010.08.031

利用 SM502 的 ZV Port 实现 SAA7111 视频采集

张斌

(中国兵器工业第五八研究所 军品部, 四川 绵阳 621000)

摘要: 以 PXA270CPU 为开发平台, 利用 SM502 的 ZVPort 实现 SAA7111 视频采集驱动开发。介绍系统的硬件和 SAA7111 驱动设计, 参照 V41 标准进行底层驱动和应用程序开发, 在设计视频设备结构实体及实例的基础上, 详细介绍了打开设备文件时的响应函数、应用层与底层驱动之间通讯的函数、图像截取函数, 实现了视频采集。该研究可为编写 Linux 驱动程序提供参考。

关键词: MMIO; V4L; ZV Port

中图分类号: TP311 **文献标识码:** A

Developing SAA7111 Video Driver Using SM502 ZV Port

Zhang Bin

(Dept. of Armament Products, No. 58 Research Institute of China Ordnance Industries, Mianyang 621000, China)

Abstract: Use the PXA270CPU as development platform, and adopt the ZVPort of SM502 to realize the SAA7111 video collection driver. Introduce the system hardware and SAA7111 driver design. Develop the bottom driver and application program according to V41 standard. Based on design video equipment entity and example, introduce the response function, communication function between applied layer and bottom layer, image interception function, realize the video collection. The research can give reference to compiling Linux driver program.

Keywords: MMIO; V4L; ZV Port

0 引言

SM502 是 Silicon Motion 公司生产的一款移动多媒体处理芯片, 具有强大的处理功能, 其中, Zoom Video Port (ZV Port) 支持与视频解码器接口, 并可直接和 NTSC/PAL 解码器 (如 SAA7111, BT891 等) 连接。故以 PXA270CPU 为开发平台, 在嵌入式 Linux-2.6.9 中利用 SM502 的 ZV Port 实现 SAA7111 视频采集驱动开发。

1 系统硬件设计

有关本系统的硬件设计, 参见 SM502 和 SAA7111 的相关数据手册, 这里就不再详述了。

2 SAA7111 驱动设计

SAA7111 是 Philips 公司生产的可编程视频处理芯片, 集 A/D 和解码功能于一身, 片内附有锁相、自动钳位、自动增益、多制式解码等电路, 可通过 I2C 总线对其工作方式进行设定。该芯片的相关驱动程序可参见嵌入式 Linux 内核 “drivers/media/video/saa7111.c” 文件, 可根据实际需要, 在 “saa7111_command” 函数中增加类似 “ioctl” 的命令。SAA7111 的 LLC、LLC2 时钟参考的产生会受到片选信号 CE 的影响, 只有当 CE 为高电平时才有效, 而在低电平时, SAA7111 将被复位。

3 SM502 ZV Port 驱动开发

3.1 Video4linux

Video4linux (V41) 是一些视频系统、视频软件、音频软件的基础, 经常使用在视频监控、webcam、可视电话等需要采集图像的场合。embedded linux 是 linux 嵌入式开发中经常使用的系统接口, 是 linux 内核提供给用户空间的编程接口。V41 分为 2 层, 底层为音视频设备在内核中的驱动, 上层为系统提供的 API, 在开发相应的驱动程序后, 就可以通过 V41 提供的系统 API 来控制视频和音频设备, 故需参照 V41 标准来实现底层驱动和应用程序开发、实现视频采集。

3.2 底层驱动开发

在最底层, Linux 系统中的每一个设备都用设备 (device) 结构的一个实例来表示, 而每一个具体的设备又在此基础上扩展出了具体的设备结构。视频设备结构实体如下:

```
struct video_device
{
    char name[32];
    int type;          /* v411 */
    int type2;        /* v412 */
    int hardware;
    int minor;
```

收稿日期: 2009-04-02; 修回日期: 2010-05-24

基金项目: “重大新药创制” 科技重大专项 (2009ZX09313-024) 资助课题

作者简介: 张斌 (1976-), 男, 四川人, 从事嵌入式 Linux 研究。

```

struct file_operations *fops;
void (*release)(struct video_device *vfd);
};
将该结构实例化为:
static struct video_device smivgx_device =
{
    .name = "PXA-SM502",
    .type = VID_TYPE_OVERLAY |
VID_TYPE_CAPTURE | VID_TYPE_SCALES |
VID_TYPE_CHROMAKEY,
    .hardware= VID_HARDWARE_SAA7111,
    .fops = &smivgx_fops,
    .release = smivgx_release,
    .minor = -1,
};
static struct file_operations smivgx_fops =
{
    .owner = THIS_MODULE,
    .open = smivgx_open,
    .release = smivgx_close,
    .ioctl = smivgx_ioctl,
    .mmap = smivgx_mmap,
};
同时, 设计驱动模块初始化和退出函数如下:
static int __init smivgx_init(void)
{
i2c_add_driver(&i2c_driver_saa7111);
if (video_register_device(&smivgx_device,
VFL_TYPE_GRABBER, -1) < 0)
{
    printk("video_register_device error\n");
    return -EINVAL;
}
return 0;
}
static void __exit smivgx_exit(void)
{
i2c_del_driver(&i2c_driver_saa7111);
video_unregister_device(&smivgx_device);
}

```

相应的函数说明为:

1) smivgx_open

使用 Open 函数打开设备文件时的响应函数, 在其中实现 SAA7111 的初始化工作: saa7111_command (&saa7111_client, DECODER_INIT, NULL)。

2) smivgx_ioctl

该函数是应用层与底层驱动之间通讯的函数, 具体的命令参见“include/linux/videodev.h”头文件。笔者主要实现的命令有: VIDIOCGCAP, VIDIOCGCHAN, VIDIOCSCHAN, VIDIOCGPICT, VIDIOCSPICT, VIDIOCGMBUF,

VIDIOC_S_INPUT, VIDIOC_S_OUTPUT, VIDIOCMCAPTURE, VIDIOCCAPTURE, 这些命令中大多设置了 SAA7111 参数和 Video4Linux 的相关参数。同时, 为了初始化 SM502 的 ZV Port 的相关寄存器, 笔者还实现了一个自定义的命令 ZV_VIDEOPANEL_PARA, 调用 SM502 显卡驱动中添加的外部函数 smivgx_set_power。具体的寄存器设置可参见 SM502 ZV Port 寄存器的相关说明。同时注意, 必须先使能 ZV Port 和它的时钟, SM502 才能使用 ZVPort 的相关功能。

3) smivgx_mmap

该函数可实现图像截取功能, 具体代码如下:
static int smivgx_mmap(struct file *file, struct vm_area_struct *vma)
{
 unsigned long start = (unsigned long)vma->vm_start;
 unsigned long end = (unsigned long)vma->vm_end;
 unsigned long size = end - start;
 unsigned long page;
 page = 0x08000000;
 if (remap_page_range(vma, start, page, size, PAGE_SHARED))
 return -EFAULT;
 return 0;
}

其中, “page = 0x08000000” 的 0x08000000 是 SM502 的 MMIO 的偏移地址, 即 PXA270 的 nCS2 决定的地址。

3.3 驱动检查

将驱动程序添加到 Linux 的内核源码体系中, 将其编译到内核中。下载内核, 重新启动后, 如果驱动没有问题, 则在/dev 目录下可以看见 video0 的设备文件, 利用 cat/dev/video0 就可以测试驱动是否正常了。

3.4 应用软件开发

遵照 V4l 的规定, 编写上层应用程序。

4 结束语

该研究在理解 Linux 驱动程序的各种体系结构的基础上实现了视频采集, 可为编写 Linux 驱动程序提供参考。

参考文献:

[1] SM502 Mobile Multimedia Companion Chip Databook.
[2] Intel PXA27x Processor Family Developer's Manual.
[3] SAA7111 Video Input Processor Databook.
[4] Video4Linux Programming.
[5] 魏永明, 耿岳, 钟书毅, 译. Linux 设备驱动程序.