
基于嵌入式 Linux 的家居监控系统设计

(作者: 龙勇)

1 系统流程架构

本系统的操作系统平台为 Linux, 内核版本为 2.6, 目标板上包括视频采集、GPRS 模块和控制器 S3C2440. 控制器主要是负责向客户显示及后端服务提供图像数据。视频服务器的启动方式是由后台或触发启动。当用户触发事件时自动启动采集程序或者由用户远程启动, 再进行图像数据的抓取。

2 系统设计

2.1 Linux 下设备驱动

Linux 是免费的操作系统, 其源代码是公开的, 我们能够根据实验的需求进行必要的研究和修改。在 Linux 操作系统中, 设备都是以文件形式的存在, 对设备的操作就是对文件的操作, 使用户可以调用通用的文件操作函数来进行访问和操作设备。

设备驱动程序是 Linux 内核与外部设备关联的接口, 它们都根据用户操作一系列的标准函数调用来执行, 通过调用独立的驱动程序来调用实际硬件的设备进行特定操作。编程接口能够使驱动程序独立运行于内核的其他部分, 就如“嵌入”到内核中一样。这种模块化设计特点使得 Linux 驱动程序的编写方法简单并且高效。操作简单的同时也有风险, 因为设备驱动程序是运行在内核中, 如果驱动程序编写出现问题, 很有可能会使整个 Linux 系统运行出现问题。

驱动程序运行与一般应用程序也有不同地方, 驱动程序运行只能调用自己相应函数和内核中的标准函数, 而一般应用程序只能调用外部数据库等应用函数。内核模块源文件中也不能包括一般的库文件, 只能使用内核中规定的内核函数, 否则就会出现错误。这也是驱动程序的写法与应用程序的写法不同之处。当然另一个不同之处是内核与一般应用程序处理错误的方式不同:

应用程序中的错误是不会对系统产生故障, 并可使用调试器来进行源代码的跟踪, 从而发现问题所在, 但内核错误有可能会影响整个系统, 即使不影响系统也会出现杀死当前进程的问题, 而且也不能使用调试来发现问题根源。

2.2 红外中断触发设计

本监控系统实现了红外热释电传感器驱动, 使用字符设备驱动。由于 S3C2440 有 117 路的 I/O 线。其 I/O 线分成 8 组端口, 分别是 GPA, GPB, GPC, GPD, GPE, GPF, GPG, GPH. 当有人进入时或离开被感应区域时, 都会使热释电红外传感器的电平发生变化, 从而引起中断。热释电红外传感器感应到有人进入时就会自动启动正在睡眠的视频采集设备, 对目标进行数据采集。同时热释电红外

传感器的驱动程序需要调用中断方式来通知内核，让内核通知视频采集器已进入工作状态，从而便于设备管理。

S3C2440 通过 GPIO 引脚的函数和宏来进行配置和控制；函数 S3C2440 _GPIO _CFGPIN 对复用 GPIO 引脚配置；函数 SET_ IRQ _ TYPE 对外部中断触发器进行设置。本监控系统使用了宏 IRQT _ BOTHEDGE 对中断进行配置。在内核头文件中定义中断号，每个外部中断都分配不同的中断号来进行控制。当内核接收到中断响应，就会使用异步通知办法。为了解决文件的异步通信，用户程序必须做到两点：第一必须给进程指定“属主”；第二为了运行异步通知，用户程序必须在设备中设置 FASYNC 标志。根据内核驱动程序写法，驱动程序必须要定义自己的 fasync 方法。当程序执行到 F_SETFL 时就启动 FASYNC，内核就会调用驱动程序 fasync. 内核空间使用异步信号来向用户发送信号，相应信号函数是：

```
Void kill_fasync (struct fasync_struct **fa, intsig, init band)
```

信号函数的功能是当红外传感器输出电平由高到低或由低到高的变化时触发中断响应，同时外部设备得到一个通过处理函数产生的电平变化，从而驱动程序中定义 read 方法。

2.3 视频采集驱动

在初始化的流程中初始化了摄像头的一些默认的值，根据红外传感器传过来的拍照命令，启动拍照程序，开始拍照，图片数据保存为 JPEG 格式存储，同时发送到手机终端 While (1)

```
{  
  
If (paizhao)  
  
{  
  
_pal=0 ;  
sendPaiZhao () ;  
readJPEG () ;  
  
keepData () ;  
  
sendData () ;  
  
}  
  
}
```

2.4 图像数据的 JPEG 压缩编码

通常情况下，没有压缩的 JPEG 图像数据占用存储空间比较大，非常耗费存储空间，也不利于网络数据传输，因此对图像数据进行压缩很有必要。对于图像的压缩方法，Linux 系统下的 Libjpeg 库中已经有实现。Libjpeg 是 Linux 下的一个标准的库，能够直接调用它来实现图片压缩。这个库的主要功能是将图片以一定的压缩比率压缩成 JPEG 文件格式的图片。由于 Linux 是开源的，这个库也是开放的。不仅可以在一般的 Linux 操作系统下工作，也可以在嵌入式 Linux 下进行工作。这样就减少了很多底层的工作，也有利于避免重复性的开发工作。

在使用这个库之前，首先需要安装和编译这个库。对于嵌入式 Linux，libjpeg 库已经包括在嵌入式 Linux 源代码包中，在配置相关库文件选项时把这个库加入，再编译一下内核就可以使用 libjpeg 库。

图像压缩封装调用 image 类的 compress 函数，如下所示：

Int image compress (image *newone, int quality) 这个函数的功能是按照指定的 quality 比率将图片压缩成 JPEG 格式。函数中的两个参数分别是指向新的 image 类数据的指针（即压缩后的图像数据），另一个参数是指定压缩比率，一般范围为 0~100. 当数字越大，压缩得到的图片质量就越高，也就越清晰，但同时压缩后的数据文件容量越大。所以压缩比不是越大越好，需要根据实验的现实需求来确定，以满足实验功能要求且存储成本和传输成本最低为标准。

2.5 短消息的收发

嵌入式监控系统通过使用串口与 GPRS 连接，通过使用 GPRS 模块来进行短消息的收发。SMS 短信的收发有两种模式，即文本模式和 PDU 模式。文本模式只能发送变通的 ASCII 字符，短信内容简单，其最大不足之处是不能收发其他字符。PDU 模式将发送的信息按照一定的格式进行新的编码，一般是十六进制数的 PDU 序列为主，把所有中文或者图像进行编码后再予以发送（如图 2-1）。

编程过程如下：

1) 打开串口

打开串口设备文件需要使用标准的 I/O 操作函数。

```
Fd=open ( “/dev/ttyS0” , O_RDWR|O_NDELAY|O_NOCTTY) ;
```

2) 设置串口属性

串口有一些规定的属性必须进行设置，否则就不能正常运行，一般要设置波特率、控制模式标志、本地模式标志、输入/输出模式标志、控制字符等相关串口属性。

3) 清空发送/接收缓冲区

为了避免以前的数据的干扰,必须对缓冲区的数据进行清空之后才能把数据读入缓冲区,这样做的目的是为了**避免不必要的**数据干扰。

4) 从串口读写数据

串口的数据读/写使用的是标准的读/写函数 (read () 和 write ())。

5) 关闭串口

关闭串口为下次再使用串口提供方便。关闭串口的函数是 close () 函数。

2.6 彩信的发送

Linux 系统中的视频子系统为 Video4Linux,他为应用程序提供了一个完整、统一的 API 函数,视频应用程序只要调用这些 API 就能完成对视频捕捉设备的操作。设备的节点文件在 #mknod/dev/videoc810

1) 打开设备

用 open () 函数来打开设备。

```
fVideo=open (“dev/video”, O_RDWR);
```

2) 获得设备参数

```
Ioctl (fVideo, VIDIOCGCAP, &vcap);
```

3) 设置图像采集参数

```
Ioctl (fVideo, VIDIOCGPICT, &vcap);
```

4) 图像采集

```
Ioctl (fVideo, VIDIOCMCAPTURE, &mem);
```

5) 等待采集结束,由于采集需要一段时间,所以应用程序需要有一个相应的等待期

```
Ioctl (fVideo, VIDIOCSYNC, &mem, frame);
```

6) 将采集的图像数据写入文件

```
Write_jpeg (buf++mb.offsets [0], mem);
```

7) 清空内存映射

Munmap (but, mb.size) ;

8) 关闭设备

Close (fVideo) ;

得到图像数据后，再把数据进行 MMS 的封装，首先是加上 MMS 头部分的数据封装，再对 SMIL 部分封装，这里主要是指指定显示 MMS 的相关信息，最后对每个输入的多媒体信息进行编码。最后通过 GPRS 模块把信息发送出去。

3 系统功能的测试结果

本文测试基于 jpeg 格式标准的视频数据的采集、传输以及显示。由摄像头完成视频数据的采集，通过由 GPRS 封装后把图像数据传输到手机。红外感应到有人进入时，摄像头取下的图像照片，并再通过 GPRS 模块把图像发到用户的手机上，从而完成了有效的监控动作。

OFweek 电子工程世界