文章编号 1000 - 5269(2011)04 - 0074 - 05

# 基于 V4L2 的 ARM11 USB 视频采集终端的设计与实现

郝 俊,孟传良*
(贵州大学 计算机科学与信息学院,贵州 贵阳 550025)

**摘 要**:文章研究用 V4L2 的架构模型开发 ARM11 - S3C6410 芯片的 USB 摄像头视频设备。首先分析 V4L2 的图像采集驱动和流程;然后在 Qt 环境下设计并实现视频采集终端软件设计,最后在友善之臂 Tiny6410 平台上移植嵌入式视频采集终端。在 S3C6410 芯片的开发板上实现实时动态采集视频图像,并且将视频传输到 PC 机上。本文对视频采集的设计方法和监控系统的后端处理具有实用价值。

**关键词**:V4L2; Qt; USB 视频采集; S3C6410; 视频监控
**中图分类号**:TP316 **文献标识码**:A

视频监控系统作为安防的重要设施被广泛应用于社会生产和生活的方方面面。

V4L2(Video for Linux2)作为 Linux 系统下的一种通用视频架构,应用于许多嵌入式视频设备之中,已形成了主流的应用模式。它为 Linux 的视频驱动提供了统一的接口,使应用程序可以使用统一的 API 函数操作不同的视频设备,极大地简化了视频及其通信系统的开发和维护[1]。

## 1 摄像头底层驱动开发与移植

图 1 所示的是标准嵌入式视频系统的一般架构。摄像头芯片采用的是中星微公司生产的 zc0303(zc3××系列芯片)。

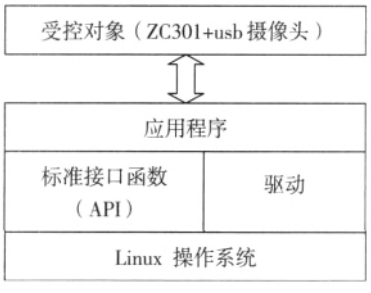


**图 1 嵌入式 Linux 系统构成**

Linux 操作系统版本选用 Linux2.6.28,系统开发步骤如下:

(1)驱动模块的注册和注销[2]

进入驱动目录/friendlyARM/linux - 2.6.28.6/drivers/media/video/ gspca 下,查看关键代码:

```
static struct usb_driver sd_driver = {        //
USB 驱动结构
    .name = MODULE_NAME,
    .id_table = device_table,
    .probe = sd_probe,
    ……
    };

    static int __init sd_mod_init(void)          //
驱动注册,初始化
    {
      if (usb_register(&sd_driver) < 0)
    return -1;
          PDEBUG(D_PROBE, "registered");
        return 0;
    };

    static void __exit sd_mod_exit(void)         //
驱动注销,退出
    {
      usb_deregister(&sd_driver);
      PDEBUG(D_PROBE, "deregistered");
    };
```

(2)配置内核

采用静态方式加载。在内核根目录下键

* 收稿日期:2011 - 07 - 04
基金项目:贵阳市工业攻关项目:税控矿产品产量监控防作弊系统(2010 筑科工合同字第 1 - 74 号)
作者简介:郝 俊(1986 - ),男,湖北随州人,硕士研究生,研究方向:嵌入式通信系统,Email:515692835@qq.com.
* 通讯作者:孟传良,Email:clmeng@gzu.edu.cn.

入 make menuconfig。主要选择如下内容,其余为默认配置。

Device Drivers - - >

Multimedia devices - - >

< * > Video For Linux

加载 video4linux 模块,为视频采集设备提供编程接口;

[ * ] Video capture adapters - - >

[ * ] V4L USB devices - - >

< * > GSPCA based webcams - - >

< * > ZC3 × × USB Camera Driver

< * > USB ZC0301 [P] Image Processor and Control Chip Support

在内核中静态加入了对采用 zc3 × × 接口芯片的 USB 数字摄像头的驱动和采集功能的支持。保存配置并且退出。

(3) 编译内核

make zImage,会在 arch/arm/boot 目录下生成 linux 内核映像文件 zImage,通过 USB 或者 SD 卡烧写到 Tiny6410 开发平台上。

(4) 查看是否配置成功

从串口终端查看主要打印输出:

usb 1 - 1.2: new full speed USB device using s3c2410 - ohci and address4

usb 1 - 1.2: Product: PC Camera

usb 1 - 1.2: Manufacturer: Vimicro Corp.

gspca: video2 created

表示配置成功,驱动可用。

## 2 视频采集操作

### 2.1 基于 V4L2 图像采集的流程

基于 V4L2 的图像采集的程序流程如图 2 所示。由于 CMOS 摄像头已经占用了设备文件 video0[6],主设备号 81,次设备号 0。所以这里创建的 USB 摄像头设备文件是 video2,主设备号 81,次设备号 2。需要提请注意的是,由于 V4L2 对 V4L 进行了彻底的改造,因此两者并不兼容[5]。

### 2.2 V4L2 的库函数与数据交换模式

V4L2 的绝大部分功能是通过 ioctl 系统调用完成的,其语法为:

ioctl( int fd,int request,void * argp)[6]

其中: fd 为设备文件描述符,通过 open( ) 函数获得;

request 为系统调用类型,用于告诉系统要做什么;

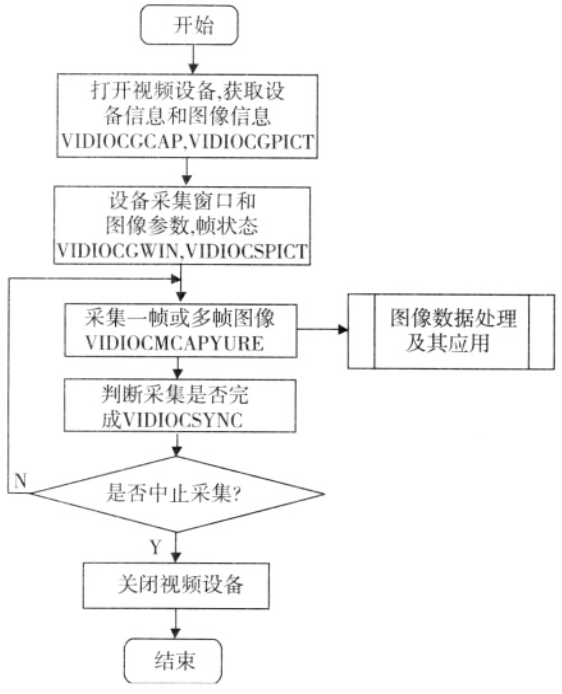


图 2　基于 V4L2 图像采集程序的流程

argp 为用户数据指针,用于传递参数或接收数据。

V4L2 支持三种数据交换模式,分别是: 直接读取设备文件方式、用户指针方式以及 mmap 映射方式[5]。本系统采用 mmap 映射方式。首先在内核空间申请多个缓冲区,然后将每个缓冲区通过 mmap 映射到用户空间,这样驱动和应用程序共享这些缓冲区,在进行数据处理时不需要进行拷贝,大大提高了效率。

### 2.3 视频采集过程

#### 2.3.1 打开视频设备

在 V4L2 中,视频设备被看做一个文件。使用 open 函数打开这个设备:

用非阻塞模式打开摄像头设备( 本系统采用)

fd = v4l2_open( dev_name, O_RDWR | O_NONBLOCK, 0);[6]

如果用阻塞模式打开摄像头设备:

fd = v4l2_open( dev_name, O_RDWR, 0);[6]

这里 dev_name 的设备文件是"/dev/video2"。

#### 2.3.2 读取设备信息

读取设备的 capability,进行分析,就知道设备具有的功能,比如是否具有视频输入,或者音频输入输出等。成功后再读取图像的属性:

```
memset( &pict, 0, sizeof( pict ) );
ioctl( fd, VIDIOCGPICT, &pict );
pict. palette = VIDEO _ PALETTE _
```

RGB24;

if( ioctl( fd, VIDIOCSPICT, &pict ) < 0 ) {

……

}[9]

### 2.3.3 截取图像的方法

mmap( ) 系统调用使得进程之间通过映射同一个普通文件实现共享内存。普通文件被映射到进程地址空间后,进程可以像访问普通内存一样对文件进行访问,不必再调用 read( ),write( ) 等操作[5]。采用共享内存通信的一个显而易见的好处就是效率高:

首先,设置 picture 的属性;

然后,初始化 video_mbuf,以得到所映射的 buffer 信息,ioctl( vd -> fd,VIDIOCGMBUF,&( vd -> mbuf) );

再,可以修改 video_mmap 和帧状态的当前设置;

最后将 mmap 与 video_mbuf 绑定。

### 2.3.4 设置视频属性

(1) 设置视频捕获格式

本系统摄像头采用的是深圳极速公司的极速普及王,捕获幅面是 800 × 600,输出格式是 RGB24,对应的结构设置如下:

fmt. type = V4L2_BUF_TYPE_VIDEO_CAPTURE;

fmt. fmt. pix. width = 800;

fmt. fmt. pix. height = 600;

fmt. fmt. pix. pixelformat = V4L2_PIX_FMT_RGB24;

xioctl( fd, VIDIOC_S_FMT, &fmt);

(2) 分配内存

视频捕获分配内存的结构体函数:

struct v4l2_requestbuffers{

__u32 count; // 缓存数量,也就是说在缓存队列里保持多少张照片

enum v4l2_buf_type type; //数据流类型,V4L2_BUF_TYPE_VIDEO_CAPTURE

enum v4l2_memory memory; // V4L2_MEMORY_MMAP 或 V4L2_MEMORY_USERPTR

};

(3) 获取并记录缓存的物理空间

首先使用 VIDIOC_REQBUFS,获取缓存数,然后通过调用 VIDIOC_QUERYBUF 命令来获取这些缓存的地址,再使用 mmap 函数转换成应用程序中的绝对地址,最后把这段缓存放入缓存队列[9]。

req. count = 2;

xioctl( fd, VIDIOC_REQBUFS, &req);

xioctl( fd, VIDIOC_QUERYBUF, &buf);

buffers [n_buffers]. start = v4l2_mmap( NULL, buf. length,

PROT_READ | PROT_WRITE, MAP_SHARED,fd, buf. m. offset);

……

### 2.3.5 开始视频采集

调用 ioctl 的 VIDIOC_STREAMON 命令开始视频采集。这里采用内存映射方式。把设备里的内存映射到应用程序中的内存控件,直接处理设备内存。

//读取缓存

xioctl( fd, VIDIOC_QUERYBUF, &buf);

//加入缓存队列

xioctl( fd, VIDIOC_QBUF, &buf);

//视频采集

xioctl( fd, VIDIOC_STREAMON, &type);

V4L2 有一个数据缓存,存放 req. count 数量的缓存数据。数据缓存采用 FIFO 的方式,当应用程序调用缓存数据时,缓存队列将最先采集到的视频数据缓存送出,并重新采集一张视频数据[9]。

### 2.3.6 结束视频采集

当视频采集活动结束的时候,需要使用 VIDIOC_STREAMOFF 命令停止视频采集,调用 close 函数关闭视频设备[5]。

//停止视频采集

xioctl( fd, VIDIOC_STREAMOFF, &type);

//关闭视频设备

v4l2_close( fd);

## 3 Qt 终端设计及其终端软件开发

### 3.1 嵌入式 Qt—Qtopia

Qt 是一个跨平台的图形用户界面库,它的核心库加入了进程间通信、多线程等模块,具有开发大规模复杂跨平台应用程序的能力[7]。Qtopia 是基于 Qt 开发的一个软件平台,主要用于嵌入式 Linux 系统的 PDA 或移动电话。Qtopia 提供了窗口操作系统、游戏和多媒体、同步框架、PIM 应用程序、Internet 应用程序、开发环境、输入法、Java 集成、本地化支持、个性化选项以及无线支持等,能够缩短开发周期。目前比较常见的有 Qtopia - 2.2.

0、Qt - Extended - 4.4.3、QtE - 4.7.0 等[7]。本系统不需要整套的桌面系统,只需单独的采集终端应用,因而选择跨平台开发性更好的 QtE - 4.7 或更高版本。

### 3.2　Qt 终端软件开发

#### 3.2.1　技术路线

本系统主要实现两大功能,一是采集视频图像,并且实时显示; 二是将采集的图像由用户选择保存到本地或是通过串口传到 PC 上。

首先编写应用框架类 QWidget,能够让视频图像窗口以及按键窗口能在程序上显示; 然后要通过 QPainter 类将通过 V4L2 采集的视频缓冲区的 Image 不断写入,做到实时显示; 最后通过信号与槽的机制编写控制按键,控制视频的显示、暂停以及退出等功能。

#### 3.2.2　Qt 的信号与槽机制的实现

信号与槽机制是 Qt 编程的基础[4],其在类型安全性和对象耦合上更胜一筹。信号是一个特定的标识,一个槽就是一个函数。当某个事件出现时,通过发送信号,可以将与之关联的槽函数激活,即执行槽函数代码。在程序中,使用 QObject::connect() 函数来关联某个信号和某个槽,而信号和槽之间的真正关联是由 Qt 的信号与槽机制来实现的。

信号与槽的关联关系一般有以下几种模式:

(1) 一个信号和一个槽关联

(2) 一个信号和多个槽关联

(3) 多个信号和一个槽关联[4]

#### 3.2.3　Qt 终端软件编写

(1) 显示区

Qt 的画图机制为屏幕显示和打印显示提供了统一的 API 接口,主要由 3 部分组成: QPainter 类、QPaintDevice 类、QPaintEngine 类[5]。其中 QPainter 类提供了丰富的操作接口,可以很方便地绘制各种图形。关键代码如下:

```
QPainter painter(this);
painter.setPen(Qt::red);
painter.drawText(rect(), Qt::AlignCenter, "贵州大学");
```

(2) 按键区

按键区利用了 Qt 的信号与槽机制,实现按键与响应函数的关联。使用基本布局类 QLayout,实现了 Qt 界面强大的布局管理功能[5]。部分代码如下:

```
//信号与槽
connect(startCapture, SIGNAL(clicked()), this, SLOT(onStartCapture()));
//布局管理
mainLayout = new QGridLayout();
mainLayout->addWidget(video,0,0);
```

(3) 其他功能区。不带窗口管理器的图像采集以及创建软件图标。

```
//图像采集
QPixmap pixmap = QPixmap::grabWidget(this->video);
//创建自定义软件图标
QWidget::setWindowIcon(QIcon(":/logo.png"));
```

软件初始化界面如图 3 所示。

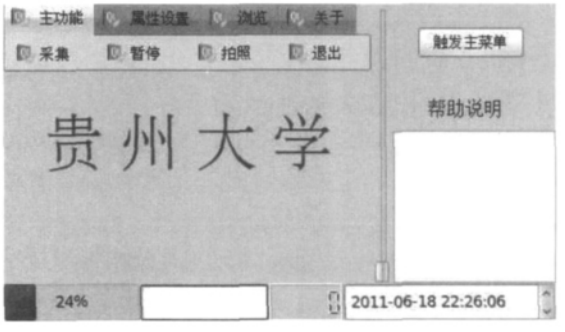


图 3　软件界面截图

## 4　ARM11 终端移植与功能实现

本项目选用 Tiny6410 核心板,它的 CPU S3C6410 基于 ARM1176JZF - S 核设计,内部集成了强大的多媒体处理单元,支持 Mpeg4,H.264/H.263 等格式的视频文件硬件编解码,可同时输出至 LCD 和 TV 显示[8]。

### 4.1　ARM11 终端移植

#### 4.1.1　交叉编译生成 ARM 平台运行的应用软件

由于编写的软件采用动态链接库的形式加载,这里需要交叉编译 libv4l 库。通过修改 Makefile 文件,添加交叉编译工具链,以生成支持 ARM 平台的二进制文件 libv4l2.so、libv4lconvert.so.

Qt 编译由 qmake 完成[4],开发软件时采用的是基于 PC 平台的 qmake,这里需要替换为 Trolltech/QtEmbedded - 4.7.0 - arm/bin/qmake,重新编译后即可生成支持 ARM 平台运行的应用软件。由于采用了动态链接库的方式,生成的软件程序非常小,非常适合于嵌入式平台的使用。

#### 4.1.2　下载到 Tiny6410 开发板

通过 SD 卡、USB 下载线或者串口终端,将上

述生成的三个文件下载到开发板上。其中 libv4l2.so、libv4lconvert.so 放到开发板的/lib 目录下,由于需要开机自动运行应用程序,这里需要将编译生成的应用软件放到/bin 目录下,并修改访问权限:

chmod +x qtCapture(其中 qtCapture 是应用程序名)

通过设置环境变量,修改/etc/init.d/rcS 文件,将/bin/qtopia & 改成:

./setqt4env(系统脚本)

qtCapture -qws &[7]

从而实现开机自动运行终端采集程序。

### 4.2 视频采集与传输

终端软件移植成功后,Tiny6410 开发板上电运行,出现如图 3 的初始化界面,点击“采集”按键,开始采集图像,并且在显示区实时动态显示。点击“拍照”,保存即时图像到根目录,可以保存到本地,也可以通过串口传输到 PC 上,作后续处理。实时采集的图像如图 4 所示。

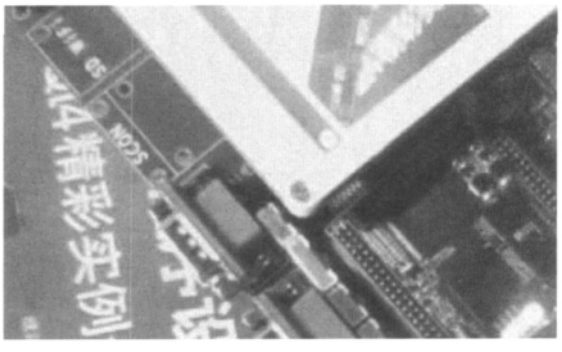

图 4 Tiny6410 开发板传回的采集图像

## 5 结束语

鉴于以往的嵌入式视频采集设备都是基于 ARM9 平台开发的,它的后续视频处理要借助于 DSP,这样的多核的处理其复杂性是可想而知的。本文创新采用 ARM11 内核的嵌入式设备,借助其强大的视频编解码功能,一体完成了后续视频处理功能,简化了开发难度,节省了成本。该平台还能够支持互联网、蓝牙、WiFi 等数据通信,可以开发出更多应用功能的视频采集监控设备。

### 参考文献:


[1] 韦东山. 嵌入式 Linux 应用开发完全手册 [M]. 北京: 人民邮电出版社,2008.

[2] 宋宝华. Linux 设备驱动开发详解 [M]. 北京: 人民邮电出版社,2006.

[3] 成洁,卢紫毅. Linux 窗口程序设计—Qt4 精彩实例分析 [M]. 北京: 清华大学出版社,2008.

[4] 闫锋欣,曾泉人,张志强译. C++ GUI Qt4 编程(第二版 [M]. 北京: 电子工业出版社,2008.

[5] 洪毅虹,曹茜. Linux 下视频监控系统的研究与设计 [J]. 电脑编程技巧与维护,2010,18:40-41,43.

[6] 张辉,李新华. 基于 V4L2 的视频设备驱动开发与移植 [J]. 电脑知识与技术,2010,6(15):3988-3990.

[7] 友善之臂. Tiny6410 用户手册 - 20110408 [EB/OL]. http://www.arm9home.net,2011.

[8] SAMSUNG Electronics. S3C6410x 32-bit RISC microprocessor user's manual (revision1.2) [EB/OL]. http://www.samsung.com,2011.

[9] Bill Dirks. Video for Linux Two API Specification: Revision 0.24. Michael H-Schimek [S].


# Design and Implement of the ARM11 USB Video Capture Terminal Based on V4L2


HAO Jun, MENG Chuan-liang*

(College of Computer Science and Information, Guizhou University, Guiyang 550025)



**Abstract:** V4L12 structural model was employed to develop USB camera video equipment of ARM11-S3C6410 chip. Firstly, the drivers and the process of V4L2 image acquisition was analyzed. Then, in the situation of Qt, the software of video capture terminal was designed and implemented. Finally, it was translplanted on the platform of Tiny6410 of FriendlyARM. Video image real-time capture was realized on the development board of S3C6410 chip and the video was transmitted to PC. There is a practical value to the design method of video capture and back-end processing of video surveillance system in this paper.



**Key words:** V4L2; Qt; USB video capture; S3C6410; video surveillance