

# 基于 Linux 及 S3C2440A 的嵌入式远程视频监控系统的的设计

吴 健,赵建军,朱继珍

(昆明理工大学 理学院,云南 昆明 650500)

**摘 要:** 介绍了一种远程视频监控系统的实现方案。其以嵌入式 Linux 和 S3C2440A 为核心平台,通过嵌入式平台将摄像头采集的视频信号进行压缩,同时进行入侵检测,再通过网络将数据传送到 Web 服务器。实现了一种体积小、成本低、数字化的监控解决方案,具有广泛的应用价值。

**关键词:** 嵌入式 Linux; S3C2440A; 图像采集; Video4Linux; JPEG 压缩

中图分类号: TP316

文献标识码: A

文章编号: 1674-7720(2011)21-0029-03

## Design of embedded remote video monitoring system based on Linux and S3C2440A

Wu Jian, Zhao Jianjun, Zhu Jizhen

(College of Science, Kunming University of Science and Technology, Kunming 650500, China)

**Abstract:** An implementation of remote video monitoring system is introduced. The system adopts embedded Linux and S3C2440A. The embedded platform can compress video signal from the USB camera, carry on the video motion detection at the same time, and then transport the signal to the web server through Internet. It realizes a small, low cost and digitized monitoring solution, which has the widespread application value.

**Key words:** embedded Linux; S3C2440A; image collecting; Video4Linux; JPEG compression

随着网络、通信和数字信息技术的不断发展,监控系统的组成模式也在快速变化和发展中,当前网络技术与嵌入式系统技术的结合催生了全新的基于嵌入式 Web 服务器的监控系统。与传统的视频采集监控系统相比,它具有可靠性高、组网方便、可远程监控等优点,因而更适用于机要部门、工厂、市场、交通运输的安防监控系统中<sup>[1]</sup>。

本文介绍了一种以 S3C2440A 开发板为基础的嵌入式远程监控系统的实现,该系统基于嵌入式 Web 服务器技术,在嵌入式硬件平台和 Linux 操作系统下进行。采用 ov511 芯片的网眼 300CMOS 摄像头进行数字图像采集,利用 JPEG 图像压缩编码方式,进行视频图像入侵检测,可以直接在以太网上解码显示。

### 1 视频监控系统总体设计

#### 1.1 系统总体结构

硬件系统是嵌入式系统的核心,它是承载软件的实

体,软件通过它来控制各种接口。本系统中硬件的总体构架包括系统存储器、外围接口电路、电源及复位电路等几个部分。系统的总体结构如图 1 所示。

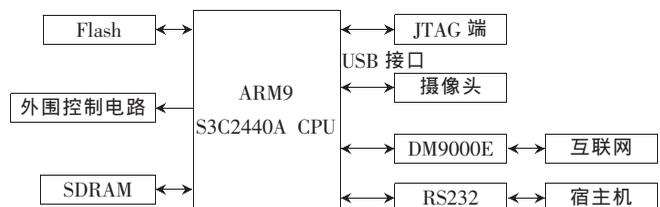


图 1 系统总体结构图

#### 1.2 S3C2440A 处理器概述

本系统使用的微处理器是三星公司生产的 S3C2440A 芯片,该芯片采用 ARM 公司的 ARM920T 的 32 bit CPU 核,并集成了 ARM 结构的 MMU 单元,各有 16 KB 的指令缓存和数据缓存,最大寻址空间为 1 GB,主频为 400 MHz (最高可达 533 Hz)。

S3C2440A 提供了丰富的片内资源:64 MB SDRAM、256 MB Nand Flash、2 MB 的 NOR Flash、LCD 控制器 (STN/TFT)、DM9000E 型网卡、4 通道的 DMA、3 通道的异步串口 (UART)、两通道的高速同步串行口 (SPI)、4 通道的带脉宽调制的 PWM 定时器和 1 通道内部定时器/看门狗定时器、双端口的 USB(主机)、1 端口的 USB(设备)、8 通道 10 bit ADC、触摸屏接口、锁相环 (PLL) 片上时钟发生器、通用 I/O 端口、相机接口、SD 卡和 MMC 卡接口。其支持各种型号的 ROM 引导 (Nor/Nand Flash, EEPROM 或其他), 1.2 V 内核供电, 1.8 V/2.5 V/3.3 V 存储器供电, 3.3 V 外部 I/O 供电, 具备 16 KB 的 I-Cache 和 16 KB D Cache/MMU 微处理器。

## 2 系统软件开发平台的建立

本系统使用 Linux 操作系统, 编译环境采用交叉编译调试方式。内核采用 Linux 2.6.30.4 版本, 使用 Cramfs 根文件系统。

### 2.1 嵌入式交叉编译环境搭建

在裁剪和定制嵌入式 Linux 之前, 必须先建立起编译环境。由于一般的嵌入式开发系统的存储空间有限, 因此通常使用交叉编译环境。简单地说, 交叉编译就是在一个平台上生成另一个平台上的可执行代码, 即在宿主机 (PC 机) 安装开发工具, 编辑、编译开发板上的引导程序 (Bootloader)、内核和根文件系统, 使其能在开发板上运行。本系统使用的编译系统为 arm-linux-gcc-3.4.1 版本。

### 2.2 Linux 系统的移植

从网上下载标准的 Linux-2.6.30.4 版本内核, 使用解压命令 #tra xvfj linux2.6.30.4tra.bz-C/opt/EmbedSky/, 然后解压到 PC 的“opt/EmbedSky”目录下。由于系统还不支持 ARM, 因此必须在系统中添加对 ARM 的支持, 进入内核源码, 修改“Makefile”文件, 将“ARCH?= (SUBARCH)”修改为“ARCH=arm”, 将“CROSS\_COMPILE?= ”修改为“CROSS\_COMPILE=arm\_linux\_”, 进行保存。在配置单中导入对内核的默认配置, 再在此基础上选择需要的功能, 如 Nand Flash、Video4Linux 编程接口函数, MTD 设备、USB 设备的支持及 Cramfs 文件系统的支持。再使用 Cramfs 制作工具 mkcramfs 把根文件目录制作成映像文件, 最后安装 DM9000E 芯片网卡的驱动程序, 即完成了系统移植。

## 3 视频数据处理模块设计

### 3.1 基于 V4L 的视频采集设计

Linux 对于视频数据采集设备的支持是通过 Video4Linux (V4L) 来实现的。V4L 是在 Linux 下用于视频和音频数据的 API 接口, 它为视频设备的应用程序提供了一系列的接口函数。这些视频设备包括市场上常见的电视捕获卡和 USB 接口的摄像头等<sup>[2]</sup>。

在编写图像采集程序时, 根据需要定义一个结构体

来保存采集过程中需要的各种参数。该结构体如下:

```
typedef struct v4l_struct{
    in fd;
    struct video_capability capability; //设备的基本信息
    struct video_channel channel[4]; //各个信号源是属性
    struct video_picture picture; //设备采集的图像的各种属性
    struct video_window window; //capture area 的信息
    struct video_capture capture;
    struct video_buffer buffer; //最底层对 buffer 的描述
    struct video_mmap mmap; //用于 mmap
    struct video_mbuf mbuf; //利用 mmap 进行映射的帧的信息
    unsigned char *map;
    int frame;
    int framestat[2];
}v4l_device;
```

从上面的结构体可以看出, 想要完成视频数据的采集, 首先要获得对应视频采集设备的信息和图像的信息, 同时需要对采集的窗口、颜色模式和帧的状态进行初始化, 然后才能进行视频图像的采集。视频采集流程如图 2 所示。

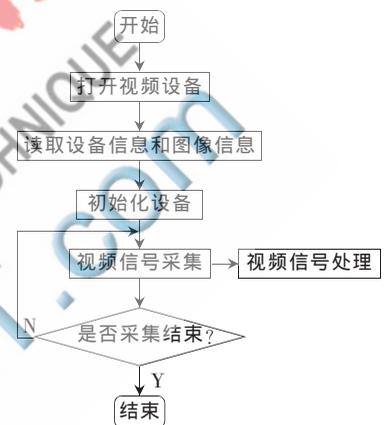


图 2 视频采集流程图

下面对 V4L 编程中使用的函数进行简单的介绍。

(1) 打开视频设备, 调用函数 int v4l\_open(char\*dev, v4l\_device\*vd); 函数调用成功后, 返回的文件描述就代表了所捕获的设备硬件。

下面的几步都会用到 ioctl() 函数来和设备进行“对话”, ioctl 是 input output control 的缩写, 函数原型是 int ioctl (int fd, ind cmd, ...)。其中, fd 表示设备的文件描述, cmd 表示用于程序对设备的控制命令, 省略号一般是一个类型的参数, 也可省略。

(2) 读取设备信息。用 ioctl () 函数读取 struct video\_capability 中有关摄像头的信息。该函数成功返回后, 将结果存放到 vd->capability 中。程序如下:

```
int v4l_get_capability(v4l_device*vd){
    if(ioctl(vd->fd, VIDIOCGCAP, &(vd->capability))<0){
        preoe("v4l_get_capability:");
        return -1;
    }
    return 0;
}
```

(3) 读取视频信息。同样使用 ioctl() 函数, 从 struct video\_picture 中读取视频信息, 函数成功返回后, 将结果存放在 vd->picture 中。调用的函数如下:

```
int v4l_get_picture(v4l_device*vd) {
```

```
if(iotcl(vd->fd, VIDIOCGPICT, &(vd->picture))<0){
    perror("v4l_get_picture:");
    return -1;
}
return 0;
}
```

(4) 视频图像截取。有两种方法截取视频图像: 直接读取设备(read())和内存映射方式(mmap())。本系统采用内存映射方式, mmap()系统调用使得进程之间通过映射同一个普通文件实现内存共享。将普通文件映射到进程的地址空间中, 进程就可以像访问普通内存一样访问文件, 无需再调用 read()、write()等操作。所调用的 mmap 代码如下:

```
int v4l_mmap_int(v4l_device *vd) {
    if(v4l_get_mbuf(vd) <0)
        return -1;
    if ((vd->map = mmap (0, vd->mbuf.size, PROT_READ |
    PROT_WRITE, MAP_SHARED, vd->fd, 0))<0){
        perror("v4l_mmap-int: mmap");
        return -1;
    }
    return 0;
}
```

执行完 mmap 之后, 便可以进行真正的图像采集, 需要调用两次 iotcl()函数, 命令代码是 VIDIOCMCAPTURE 和 VIDIOCSYNC。VIDIOCMCAPTURE 的作用是告知 iotcl()将图像数据采集到 mmap 所映射的内存中。如果调用成功, 就开始一帧图像的截取, VIDIOCSYNC 用来判断这一帧的截取是否成功, 若成功, 就表明这一帧的截取已完成, 可以开始下一帧的截取。本系统采用连续帧采集方式, 具体的代码就不在这赘述了<sup>[3]</sup>。

### 3.2 视频数据压缩

在获得视频数据后, 由于原始的图像数据量较大, 网络带宽有限, 需要在网络传输前进行压缩。本系统由于硬件条件的限制, 为了达到远程视频监控的效果, 采用基于 MJPEG 算法进行视频压缩。其主要特点是动态使用 JPEG 算法, 基本不考虑视频流中不同帧之间的变化, 只单独对某一帧进行 JPEG 压缩, 配合嵌入式 Web 服务器, 采用基于 Socket 的编程, 实现了面向用户端的视频监控<sup>[4]</sup>。

对于 Linux 下的 JPEG 图像数据压缩, 可以使用 Libjpeg 库实现。Libjpeg 是 Linux 下的一个标准而常用的库, 它的功能是将图片以一定的压缩比率压缩成如 JPEG 格式的图片, 或者对 JPEG 图片进行解压缩以及其他一些对 JPEG 图片进行处理的功能。Libjpeg 的主要文件有 jpeglib.h、libjpeg.a 和 libjpeg.so 等。可以去网上下载 Libjpeg 的源码, 取得文件 jpegsrc.v6b.tar.gz, 放于/usr/src 目录下。依次执行:

```
#cd/usr/src
```

```
#tar xzvf jpegsrc.v6b.tar.gz
#./configure
#make
#make install
```

执行完上述命令后, jpeglib.h 被拷到/usr/include 目录下, libjpeg.a 和 libjpeg.so 被拷到/usr/local/lib 目录下, 至此, Libjpeg 库的安装配置完成。

### 3.3 视频图像的入侵检测

视频序列检测是为了能够实现在监控过程中的自动报警。报警系统是视频监控系统中不可或缺的一部分。在数字视频监控系统中, 图像序列的运动检测及报警不仅可以自行替代监视人员的部分工作, 提高监视系统的自动化水平, 而且还可以提高监控存储效率。

本视频监控系统主要是对视频是否有入侵对象进行分析检测, 一旦画面上出现超过阈值的变化就会自动报警。

运动目标检测的方法主要可以分为帧差法、流光法和背景差法三种。本系统使用背景差法, 因为视频监控主要使用规定的摄像机对场景进行监控, 场景固定。背景差法的基本思想是通过输入图像与背景模型进行比较的方法检测运动目标<sup>[5]</sup>。

本文结合视频监控系统发展的方向, 给出了一种基于嵌入式 ARM 的视频监控系统设计方案, 并给出了系统的实现方法。本系统采用了基于模块的设计方法, 各个模块之间相互独立, 增强了系统的健壮性和灵活性, 当需要更换其中一个模块时, 其他模块并不需要进行很大的改动, 有利于系统的更新换代。

#### 参考文献

- [1] 于明, 范书瑞, 普祥焯. ARM9 嵌入式系统设计与开发教程[M]. 北京: 电子工业出版社, 2006.
- [2] 张晓东, 李秀娟, 张杰. 基于 ARM 的嵌入式远程监控系统设计[J]. 现代电子技术, 2008, 31(6): 22-23.
- [3] 杨颖, 陈之龙, 黄志. 基于 USB 摄像头的嵌入式远程视频监控[J]. 安防科技, 2007(3): 24-26.
- [4] 苏日建, 宋胜利. 嵌入式图像采集系统的 JPEG 算法改进[J]. 重庆工学院学报, 2006, 20(11): 75-76.
- [5] 付思华, 张小虎. 基于序列图像的运动目标实时检测方法[J]. 光学技术, 2004, 30(2): 215-217.

(收稿日期: 2011-06-01)

#### 作者简介:

吴健, 男, 1987 年生, 硕士研究生, 主要研究方向: 嵌入式系统。

赵建军, 男, 1962 年生, 副教授, 硕士生导师, 主要研究方向: 电子技术及嵌入式系统。

朱继珍, 男, 1986 年生, 硕士研究生, 主要研究方向: 嵌入式系统。