

基于电力载波视频监控系统的软件设计

秦 瑶,田逢春,李 川,韩 亮,王姗姗
(重庆大学 通信工程学院,重庆 400044)

摘 要:随着电力线通信的广泛应用,电力载波视频监控也越来越受到重视。文中提出一种基于电力载波的视频监控系统的软件设计的方法。系统采用 ARM+Linux 的软硬件平台,将摄像头采集来的视频图像经过压缩芯片压缩后,再通过实时传输协议 RTP/RTCP 和传输层的 UDP 协议进行传输,然后在 PC 机上接收数据并显示出来。整个系统建立在嵌入式 Linux 平台上,能独立完成视频的采集、压缩和传输功能,可广泛应用于视频监控领域。实验证明该方法是可行的。

关键词:嵌入式 Linux; 视频监控系统; 软件设计

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2009)09-0005-04

Software Design of Video Monitoring System on Power Line

QIN Yao, TIAN Feng-chun, LI Chuan, HAN Liang, WANG Shan-shan

(College of Communication Engineering of Chongqing University, Chongqing 400044, China)

Abstract: With the extensive application of power line communication, the video surveillance on power line carrier has become increasingly attention. Introduces the software design of a video monitoring system based on power line. With the ARM and Linux platform, the collected video images from cameras are compressed by compression chip. Then the images are transmitted through the real-time transport protocol RTP/RTCP and the transport layer UDP protocol. At last the data are received and shown on a computer. The whole system is established on an embedded Linux platform with the function of video images collection, compression and transmission. It can be generalized to the application of most video monitoring area. Experiments show that the method is feasible.

Key words: embedded Linux; video monitoring system; software design

0 引言

随着计算机技术及网络技术的迅速发展,在银行、商场、城市交通、酒店等安全保卫方面,视频监控起着重要的作用;然而为了保障人们自身安全,在家庭和小区进行监控也是很有必要的。目前,视频监控系统大多是采用嵌入式网络摄像机,利用网络进行传输,利用嵌入式监控设备体积小、性能稳定、通讯便利等特点^[1]。针对我国互联网的现状,综合采用各种接入方式,老百姓身边四通八达的电力网就提供了一个较为理想的宽带接入解决方案。目前,电力载波通信(PLC, Power Line Communication)已较为成熟,典型的应用是水、电、气表抄表系统。在电力数据通信组网方面虽然还没有应用得这么普遍,但国际上已有相关标准,如为高速电力线网络产品和服务制订国际规格的

产业组织—家庭电力线联盟(HomePlug Powerline Alliance, HPA)制定的 HomePlug AV 规范中的传输速度已达到 200Mbps,主要应用于家庭数字化中的组网等^[2]。此外,国外有关公司也生产出了相应的电力载波通信芯片和电力载波组网的产品。

基于这种发展趋势,笔者提出了一种基于电力载波的视频监控系统,该系统由用户终端和计算机监控中心组成。用户终端把视频图像进行采集压缩处理后,由主控系统封装然后发给 PLC 芯片,PLC 芯片再封装成 PLC 数据帧发送到电力线上进行传输,接收时还需先经过 PLC 芯片拆包后再传给计算机,计算机接收并进行解压处理,最终播放视频。电力载波视频监控的主要优点体现在分布范围广,不用重新布线或改造,能实现即插即用。一个典型的电力载波视频监控系统组网如图 1 所示。

1 视频监控系统总体设计

系统的总体设计包括硬件设计和软件设计。系统的总体结构图如图 2 所示。

收稿日期:2009-02-10;修回日期:2009-04-13

基金项目:国家大学生创新性计划项目(081061104)

作者简介:秦 瑶(1984-),女,四川成都人,硕士研究生,从事电力载波通信方面的研究;田逢春,教授,博士生导师,研究方向为通信技术、信息光学、图像处理、生物信息学。

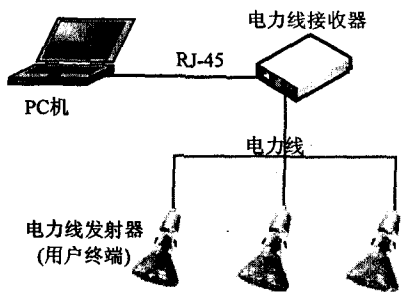


图 1 电力载波视频监控系统组网图

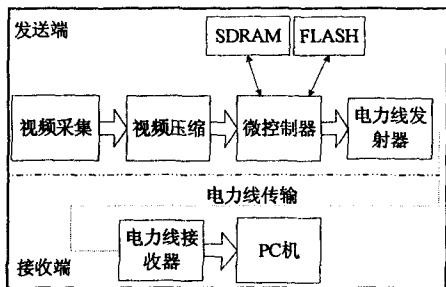


图 2 系统总体结构图

硬件设计部分主要包括视频采集及压缩编码模块、微控制器模块、电力收发器模块、设备驱动模块 4 部分。微控制器主要由主控芯片、Flash 和 SDRAM 组成,其中,主控芯片是整个控制系统的核心,它负责整个系统的调度工作。当系统启动时,微控制器用 Boot-Loader 将 Linux 内核加载到 SDRAM 中, Linux 操作系统从 SDRAM 中启动^[3]。系统启动后,微控制器控制视频采集模块采集数据,并将视频数据流转换成 PLC 芯片的数据帧格式,再送往 PLC 芯片打包调制。

软件设计部分主要包括嵌入式 Linux 开发环境的建立、视频采集与压缩编码模块、通信模块 3 部分。嵌入式 Linux 系统负责整个系统软件资源分配和任务调度工作。压缩编码模块主要负责将采集到的视频流压缩为 MPEG-4 的数据流。通信模块主要负责 MPEG-4 视频流的传输与控制。系统软件流程图如图 3 所示。

2 系统软件设计和实现

2.1 嵌入式 Linux 开发环境

实时操作系统(RTOS)是嵌入式应用软件的基础和开发平台,应用程序都是建立在它之上。实时嵌入式操作系统的种类繁多,大体上可分为两种:商用型和免费型。商用型的实时操作系统功能稳定、可靠,有完善的技术支持和售后服务,但价格昂贵。免费型的实时操作系统在价格方面具有优势,目前主要有 Linux、WinCE 等^[4]。

与其它嵌入式操作系统相比,嵌入式 Linux 具有开放源代码、高可靠性以及强大的网络功能等优势,因

此选用嵌入式 Linux 作为系统的软件平台。

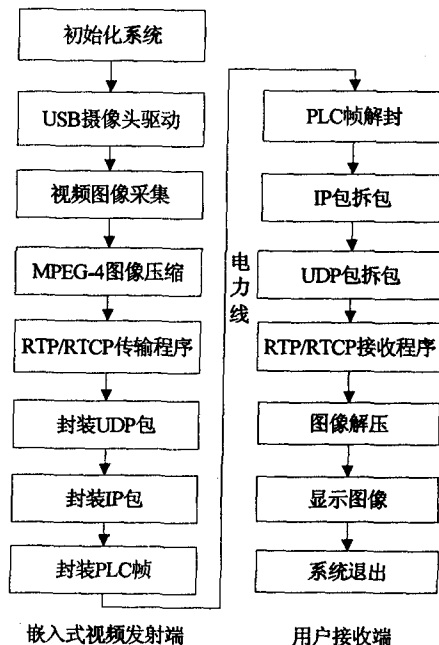


图 3 系统软件流程图

通常的嵌入式系统软件开发采用交叉编译调试的方式。交叉编译调试环境建立在宿主机(即 PC 机)上,对应的开发板叫做目标板(即 ARM 嵌入式系统)。由于宿主机和目标板上的处理器不同,宿主机通常为 Intel 处理器,而目标板为 ARM 处理器,所以需要交叉编译工具生成在对应平台上可运行的代码。在本系统中,需要在宿主机 PC 上安装 Linux 系统,并通过修改 Linux 内核的 makefile 文件,对 Linux 内核进行相应的配置。然后,在宿主机上安装 GNU 交叉编译工具,从而建立交叉编译调试的开发环境。文中采用移植性很强的 C 语言在宿主机上编写视频传输程序,再利用交叉编译调试工具 arm - linux - gcc 编译链接生成可执行代码,最后向目标平台移植。

2.2 视频采集与压缩模块

视频采集模块的核心部分是 V4L,即 video4linux,是 Linux 内核中的一类 API,它提供了应用程序所需要的一系列接口函数,通过这些函数可以执行打开、读写和关闭等基本操作^[5]。再配上相应的视频采集设备,就可以实现视频图像的采集。

视频采集流程如图 4 所示。首先打开摄像头设备,然后调用 v4l_set_picture()对摄像头的分辨率、图像格式等进行设置。在 V4L 中有两种视频采集方式,一种是调用 v4l_read()直接读取设备的方式,另一种是调用 v4l_mmap()通过内存映射的方式读取视频。本系统采用第二种方式进行视频采集。

视频压缩是提高系统性能的重要环节。本系统采用 MPEG-4 的压缩芯片对采集到的视频进行压缩编

码。采集压缩后的视频数据放在内存中以便传输。

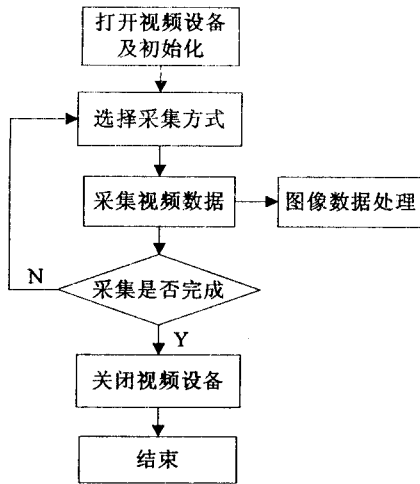


图4 视频采集流程图

2.3 通信模块设计

由于视频数据具有数据量大、允许有一定误码率等特点,对 MPEG-4 压缩后的视频流采用重实时性而非完整性的 UDP 协议传输;同时为了保证可靠的传输质量,通信模块采用了实时传输协议 RTP/RTCP^[6]。该协议建立在传统的 TCP/IP 协议之上,能够提供流媒体数据高效率传输,同时也对多方用户连接建立、网络状况实时侦测、用户管理、网络异常处理等有很好的支持。

网络通信模块是系统的主体部分,其设计开发是通过网络编程接口(Socket)来实现的。按照系统客户/服务器(C/S)的网络传输模型,两端通过网络建立 TCP/IP 连接,按照自定义的数据通信协议交换数据,完成数据通信和系统控制功能。

文中采用数据报套接字(SOCK_DGRAM),它是基于 UDP 协议的一种套接字,是一种无连接的服务,数据通过相互独立的报文进行传输,是无序的,并且不保证可靠和无差错。但为了保证程序的正常工作,每个使用 UDP 的程序都要有自己的对数据进行确认的协议^[7]。UDP 视频传输流程如图 5 所示。本系统中,在 ARM 开发板上移植 Web 服务器,把 PC 机作为客户端接收数据。

2.3.1 Socket 编程原理

Socket 是进程间通信的接口,即套接字。它可以用于同一台计算机应用层与传输层之间的通信;也可以用于一台计算机的某个进程与另一台计算机的某个进程的通信,即跨网络进行。

一个完整的 Socket 用一个相关描述{协议,本地地址,本地端口,远程地址,远程端口}

每一个 Socket 有一个本地的唯一 Socket 号,由操作系统分配。

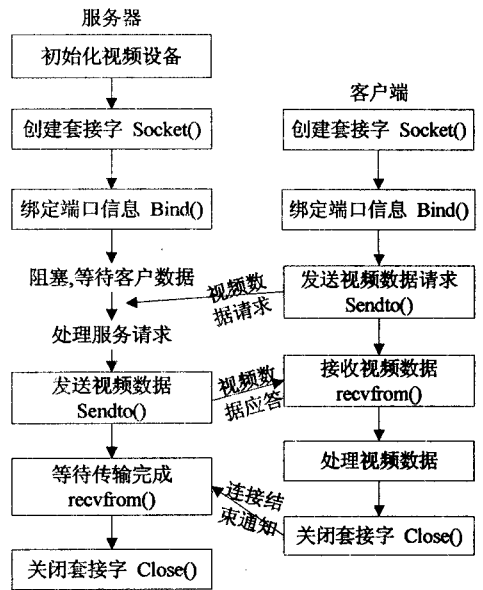


图5 UDP 视频传输流程图

Socket 有三种类型:流式套接字(SOCK_STREAM),数据报套接字(SOCK_DGRAM)及原始套接字。文中所用到的是数据报套接字,其中涉及到的函数原型如下^[8]:

1)int socket(int domain,int type,int protocol)。参数 domain 设置 AF_INET,针对 Internet 在远程主机之间通信。参数 type 指明创建的套接字类型,对应的参数值为 SOCK_DGRAM:数据报套接字,表明用的是 UDP 协议。参数 protocol 一般缺省为 0。

2)int bind(int sockfd,struct sockaddr * my_addr,int addrlen)。指定一个套接字使用的端口。my_addr 是一个指向 struct sockaddr 的指针,包含有关地址的信息:名称、端口和 IP 地址。Addrlen 可以设置为 sizeof(struct sockaddr)。

3)int sendto(int sockfd,onst void * msg,int len,unsigned int flags,struct sockaddr * to,int tolen)和 int recvfrom(int sockfd,void * buf,int len,unsigned int flags,struct sockaddr * from,int tolen)。这两个函数是进行无连接的 UDP 通讯时使用的。使用这两个函数,则数据会在没有建立过任何连接的网路上传输。

4)int close(int sockfd)。当程序进行网络传输完毕后,需要关闭这个套接字。执行 close()之后,套接字将不会再允许进行读操作和写操作。

2.3.2 服务器端设计

一般网络监控系统可以基于 C/S 或 B/S 架构进行设计。本系统主要是在局域网中实现实时监控功能,将采用 C/S 架构,即 Client/Server(客户机/服务器)结构^[9]来实现。服务器端在前端 ARM 开发板上建立 Web 服务器,等待收到客户端的数据请求后再发

送数据。

在设计服务器端时,采用了多线程的设计方法,将视频采集与压缩和通信模块整合成一个并发式多线程服务的嵌入式视频服务器。这个多线程机制中,主线程负责初始化视频采集设备,从摄像头硬件中读取视频数据,并包含部分编码工作,把处理后的视频数据保存到共享内存区。而从线程则是负责将共享内存区的视频码流通过通信模块发送到客户端。从线程工作时不会影响主线程的运行。这样有利于提高应用程序的实时性,充分利用系统资源。

视频传输中,在服务器端要求视频采集和发送互不干扰,避免采集和发送相互等待或相互冲突的情况出现。对于多线程的共享内存分配,本系统采用了一种基于环形缓冲区的多线程视频传输技术,较好地解决了这个问题。共享内存环形缓冲区的数设置为 4,即四个缓冲区,主线程对这四个缓冲区的访问,即把视频采集压缩后的数据存放到共享的缓冲区中,同样从线程的访问即是共享缓冲区的数据发送给 PLC 芯片,再由 PLC 芯片对数据进行处理后送到电力线上传输。无论是哪个线程的访问,对四个缓冲区都是顺序访问的,然而线程间的访问应该是互斥的,在某一时刻只有一个线程对某个共享缓冲进行读写访问。

主从线程的具体实现:初始化以后,主线程即视频采集线程进入一个循环,由于使用 UDP 协议是无连接的,所以在端口绑定后线程阻塞,等待一个连接的到来。在连接到来时候,主线程继续执行,调用线程动态创建函数打开一个从线程,即数据传输线程,将共享内存中的视频数据发送到客户端。主线程再次被阻塞等待新的数据连接的到来。其伪代码如下:

```
int video_init(vd, video_fd, format, width, height, method); /*
初始化摄像头 */
pthread_attr_t attr; /* 初始化共享内存区 */
pthread_mutex_t mutex; /* 初始化主线程参数及互斥 */
sockfd = socket(AF_INET, SOCK_STREAM, 0) /* 打开套接口 */
pthread_create(&thread, &attr, video_send, sockfd, video_fd, format, width, height, method); /* 创建视频循环采集主线程 */
while(1) {
    if(bind(sockfd, (struct sockaddr *)&servaddr, sizeof(servaddr)) = -1)
    /* 检查设定端口绑定 */
        { perror("bind error");
        exit(-1); }
    /* 检查客户端是否连接上,如果已连接则创建从线程以发送视频数据 */
    pthread_create(&thread, &attr, video_send, sockfd, video_fd, format, width, height, method); /* 创建视频循环发送从线程 */
    ...
}
```

2.3.3 客户端设计

本系统的开发是在 Linux 主机上进行,所以为了方便设计,客户端也在 Linux 下进行设计。需要在 PC 机上建立 ARM-Linux-gcc 交叉编译环境。

要与服务器端建立 Socket 连接,客户端首先要创建一个 Socket,用 Socket 函数来建立,仍然建立流式套接字。然后需要检查一下用户所输入的服务器的地址和端口是否正确,代码如下:

```
if(sockfd = socket(AF_INET, SOCK_STREAM, 0) = -1)
{ printf("Open client socket error! \n");
exit(1); }
```

初始化好 Socket 和服务器端的地址和端口,接着客户端主动向服务器端请求数据,服务器收到请求后,服务器端将采集压缩后的数据通过 Socket 传送给客户端,客户端接收压缩数据,将图像数据解压并显示在屏幕上。

3 结束语

根据目前嵌入式系统、网络的发展,考虑到系统硬件设备,文中设计了基于电力载波视频监控系统的软件部分。使用 ARM-Linux 开发平台,主要设计了视频的通信传输模块。视频图像经过采集压缩后,使用 RTP/RTCP 实时传输协议,通过 UDP 打包后由电力线传输给计算机。实验证明这种方法可行,并且效果较好。

参考文献:

- [1] 王刚,赵跃龙.基于嵌入式网络视频监控系统的设计[J].湖南学院学报,2006,27(5):37-40.
- [2] Zeng Xu, Liang Ming, Liu Huaizhi, et al. Monitoring System of Distribution Running States Based on Broadband Power Line Communication[C]//International Conference on Power System Technology. [s.l.]:[s.n.], 2006:1-6.
- [3] 蔡士义,冀小平.嵌入式视频监控系统的设计与实现[J].科技情报开发与经济,2007,17(32):198-199.
- [4] 李霏,王让定,徐霏.基于嵌入式 Linux 的多路视频监控系统的的设计[J].宁波大学学报:理工版,2007,20(2):151-154.
- [5] Zhao Jichun, Sun Sufen, Yu Feng, et al. The design and realization of embedded wireless video monitoring system based on GPRS[J]. Wireless Communications, Networking and Mobile Computing, 2008(10):1-4.
- [6] 操龙敏,蒋建国,齐美彬. RTP 协议在嵌入式网络摄像机中的设计与实现[J].计算机技术与发展,2008,18(3):214-216.

(下转第 12 页)

此,针对电影视频来说 $m = 0.6$ 时效果较好。

表 1 实验结果数据

实验视频	总场景数	总帧数	时长	正确检出数	漏检数	误检数	查全率	查准率
樱	21	2879	2min	18	3	0	85.7%	100%
男	14	2161	1min30s	12	2	2	85.7%	85.7%
超	4	2495	1min43s	4	1	1	80%	80%
变	9	2471	1min42s	9	1	3	90%	75%
长	25	3215	2min13s	34	1	10	56%	60%

注:樱:樱桃小丸子;男:男儿本色;超:超人特工队;变:变相怪杰;长:长江七号。

从表 1 中可以看出,查全率和查准率都超过了 80%,这说明了基于 HSV 颜色空间模型的场景切分算法取得了很不错的效果。对于不同的 AVI 视频文件,查全率和查准率可能会相差比较大,原因就是阈值 m 的取值影响了场景切分的准确性与完全性。

图 2 的两幅图像是 NBA 球赛视频中相邻的两帧,而在本实验中被切分在不同的子视频里。图 3 的两帧图像在《生化危机》视频中属于不同的场景,却没有被切分出来。出现这样的情况就是因为 m 的取值造成的。进一步实验证明,当 $m = 0.7$ 的时候,图 3 中的两帧图像就被切分到了不同的子视频中。



图 2 NBA 比赛视频



图 3 生化危机视频截图

5 结束语

基于 HSV 空间颜色模型的场景切分算法,改进了以往的基于 RGB 颜色模型的灰度直方图的场景切分

算法,首先将图像的 RGB 值转换为 HSV 值,然后量化得出一维特征矢量值,再计算其直方图,通过统计比较计算得出两幅图像的相似性,最后根据其相似性与实验给定的阈值进行比较,以判断是否发生了场景切换。这种算法避免了两幅不同图像之间因 RGB 颜色直方图却有可能相同而无法区分的缺点,从而更加准确地对视频进行场景切分。经过实验证明,这种基于 HSV 空间颜色模型的场景切分算法在对 AVI 视频文件进行场景切分取得了很好的效果,能够比较准确地分割出合适的子视频,达到了预期的目的,为后部分的视频关键帧的提取提供了基本的技术条件。后续的工作将是对渐变镜头进行检测,使得场景的切分更加准确。同时,由于不同场景之间的音频也会不同,因此,检测音频的切换也可以检测出场景是否发生了切分。这也是后续的工作重点。

参考文献:

- [1] Lienhart R, Pfeiffer S, Effelsberg W. Video Abstracting[J]. Communications of ACM, 1997, 40(12): 54 - 62.
- [2] Yeung M, Yeo B L, Liu B. Extracting Story Units form Long Programs for Video Browsing and Navigation. Proc [C]// IEEE Multimedia Computing & Systems 1996. Hiroshima, Japan: [s. n.], 1996: 296 - 305.
- [3] Winnemoller H, Olsen S C, Gooch B. Real - Time Video Abstraction[J]. ACM Transactions on Graphics (TOG), 2006, 25(3): 1221 - 1226.
- [4] 高俊杰. 一种基于交互信息量的视频摘要生成方法[J]. 微电子学与计算机, 2007, 24(2): 128 - 131.
- [5] 姚洪英, 范铁生. 基于压缩域的视频摘要技术的研究[J]. 鞍山师范学院学报, 2004, 6(2): 67 - 68.
- [6] 苏新宁. 视频信息索引技术研究进展[J]. 情报学报, 2004, 23(4): 440 - 446.
- [7] 程文刚, 须 德. 一种层次视频摘要生成方法[J]. 中国图象图形学报, 2004, 9(1): 118 - 123.
- [8] Miyahara M, Yasuhida Y. Mathematical transform of (R, G, B) color data to Munsell(H, V, C) color data[C]//In: Visual Communications and Image Processing '88. Cambridge: [s. n.], 1988: 650 - 657.
- [9] 王 峰, 刘文波. 基于频域的指纹图像增强算法[J]. 计算机技术与发展, 2008, 18(7): 12 - 16.

(上接第 8 页)

- [7] 张素文, 付 薇, 刘明兰. 嵌入式视频图像传输系统的设计与实现[J]. 微计算机信息, 2007(3 - 2): 60 - 62.
- [8] Matthew N, Stones R. Linux 程序设计[M]. 第 2 版. 杨晓云, 王建桥, 杨 涛, 高文雅等, 译. 北京: 机械工业出版社,

2002.

- [9] Li Jing, Hao Weidong. Research and Design of Embedded Network Video Monitoring System Based on Linux[J]. Computer Science and Software Engineering, 2008(9): 1310 - 1313.