

嵌入式网络摄像机的设计与实现

郑鹭斌, 纪景清, 陈华云, 朱英杰

(厦门理工学院, 福建 厦门 361024)

摘要:具有网络功能、便携式、数字化、可视化的视频监控系统已经成为安全技术领域的重要组成部分。系统设计和实现了一种基于 Intel PXA270 硬件核心、Linux 软件平台的网络摄像机。使用 USB 接口摄像头来采集视频数据,并且利用流媒体技术实现多客户端的实时显示。嵌入式网络摄像机拥有独立的 IP 地址,无需与 PC 机连接,可以安置在任何一个具备 IP 网络接口的地点。远端用户可在自己的 PC 上使用标准的网络浏览器,对网络摄像机进行访问,实时地监控目标现场的情况。

关键词:PXA 270; Linux; 流媒体; 摄像机

中图分类号:TP39

文献标识码:A

文章编号:1673-629X(2012)01-0168-04

Design and Implementation of Web Camera Based on Embedded System

ZHENG Lu-bin, JI Jing-qing, CHEN Hua-yun, ZHU Ying-jie

(Xiamen University of Technology, Xiamen 361024, China)

Abstract: The video monitoring systems including network, digital technology, visualization technology have been the most important part of security technology. The web camera is designed and implemented based on Intel PXA270 core and Linux in this system which uses USB interface camera to collect video and streaming media technology to realize multi-user display. The web camera which owns IP is able to work anywhere without any PC except network. Customers can monitor and control remote target using the camera by standard IE.

Key words: PXA 270; Linux; streaming media; camera

0 引言

随着科技的发展,安全技术已经成为各行各业的重点对象,具有网络功能、便携式、数字化、可视化的视频监控系统已经成为安全领域的重要组成部分^[1]。嵌入式网络摄像机是网络摄像机、传统摄像机与网络视频技术相结合的新一代产品,是集嵌入式技术、视频压缩技术、计算机技术、网络技术等多种先进技术于一体的数字摄像设备^[2,3]。基于网络的实时视频采集、处理、传输系统,利用流媒体技术实现多客户端的实时显示,具有不受距离限制、廉价、通用的特点,极大提高了系统的应用性。

以网络摄像机作为模型,对计算机视觉、机器人技术、远程教学、安全监控等领域都有极其重要的意义^[4]。

1 系统设计

1.1 硬件选型

随着嵌入式系统的高速发展和普及应用,人们在系统设计中不再单纯考虑产品功能了,其功耗也成为系统设计中所需考虑的重要环节。如今,仅是功能丰富而电量消耗过盛的嵌入式产品已不再满足人们的需求^[5]。PXA270 处理器,作为一款优秀的嵌入式处理器,配合嵌入式 Linux 或 Wince 操作系统,理论上可以支持任何媒体格式,是英特尔公司于 2003 年底推出的性能最为强劲的 PXA27x 系列嵌入式处理器。PXA27x 系列嵌入式处理器基于 ARMv5E 的 Xscale 核心,最高频率可达 624MHz。PXA270 处理器集成了存储单元控制器、电源控制器、时钟、LCD 控制器、DMA 控制器、I2S 控制器、AC97 控制器、FIR 控制器等众多外围控制器,提供了丰富的外围接口,以供多种功能模块的实现^[6]。其 wireless MMX 技术和 SpeedStep 动态电源管理技术可以很有效地降低系统电源功耗。

基于以上优点,使得 PXA270 得以广泛应用于 PDA、智能手机、医疗器械、远程通信、Web 记事本等众

收稿日期:2011-06-14;修回日期:2011-09-25

基金项目:福建省教育科技计划项目(JB09199)

作者简介:郑鹭斌(1975-),男,工程师,研究方向为嵌入式系统、计算机网络。

多领域^[7]。PXA270 处理器拥有的丰富硬件资源为网络摄像机的设计提供了方便的硬件基础,因此本系统采用 Intel PXA270 作为硬件核心。

本系统的视频信号采集选取 USB 摄像头,源于摄像头驱动程序较为丰富,易于系统实现且节约 CPU 资源。本系统采用的是中微星 ZC301 摄像头。

1.2 软件系统设计

整个系统以嵌入式 Linux 为基础构建而成,可以把系统分成两层:系统层和应用层。系统层模块主要工作包括 Linux 内核的剪裁和移植、摄像头驱动程序接口设计。应用层功能主要实现图像的采集与压缩、视频网络的传输和视频的实时显示等。系统软件结构如图 1 所示。

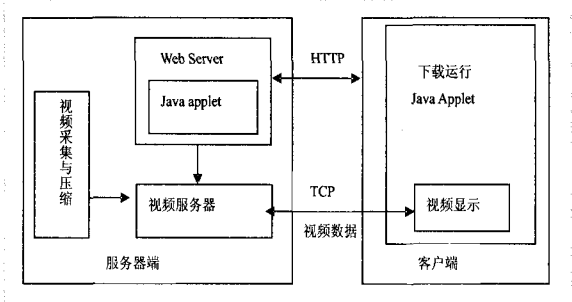


图 1 系统软件结构

2 模块设计与实现

2.1 摄像头驱动移植

通过已知的摄像头型号,可直接下载相应的驱动程序,亦可使用万能驱动程序 gspcav1-20071224^[8]。将它解压到内核的相应目录(一般是 drivers/usb/media 目录下,如若不存在,用户可自行创建)。然后修改 drivers/usb 目录下的 Makefile 和 Kconfig 文件,用以加载 gspcav 的驱动编译选项,最后再编译进内核即可。Linux 2.6 以上的版本目录为:

- 1) Device Drivers \ Multimedia devices \ Video For Linux
- 2) Device Drivers \ USB support \ Support for Host-side USB \ USB Host Controller Drivers \ OHCI HCD support
- 3) USB Multimedia devices \ USB SPCA5XX Sunplus \ Vimicro \ Sonix jpeg Cameras

将新生成的内核 zImage 烧写入开发板,并插入 USB 摄像头,摄像头即可识别图像。设备名所在路径一般为 /dev/video0。

2.2 图像的采集与压缩

视频采集模块是嵌入式网络摄像机的核心模块之一。它作为 Linux Kernel 中提供的一组支持音视频的 APIs,通过 Linux 操作系统调用影像设备驱动程序和

V4L(ideo4linux)来捕获视频^[9]。

由于摄像头设备文件所在的路径通常为 /dev/video0,因此通过对 V4L APIs 的调用,读取 /dev/video0 设备文件,即可实现对摄像头捕获的视频数据进行采集,其具体流程如下:

- (1) 打开视频设备; // int v4l_open(char *, v4l_device *)
- (2) 获得设备信息; // int v4l_get_capability(v4l_device *)
- (3) 根据需要更改设备的相关设置;
- (4) 获得采集到的图像数据(v4l 提供了两种方式:直接通过打开的设备读取数据或使用 mmap 内存映射的方式获取数据); // int v4l_get_picture(v4l_device * vd)
- (5) 对采集到的数据进行操作(如显示到屏幕,图像处理,存储成图片文件);
- (6) 关闭视频设备。 // int v4l_close(v4l_device * vd)

v4l_device 设备结构定义如下:

```
struct _v4l_struct
{
    int fd; //保存打开视频文件的设备描述符;
    struct video_capability capability; //该结构及下面的结构为 v4l 所定义可在上述头文件找到
    struct video_picture picture; //采集到的图像的属性;
    struct video_mmap mmap;
    struct video_mbuf mbuf; //利用 mmap 进行映射的帧的信息;
    unsigned char * map; //用于指向图像数据的指针;
    int frame_current;
    int frame_using[ VIDEO_MAXFRAME]; //这两个变量用于双缓冲;
};
typedef struct _v4l_struct v4l_device;
```

截取的图像数据的方法:

- 1) 使用 mmap 内存映射的方式获取视频数据: mmap 是一个内存映射函数,它把文件内容复制到一段虚拟内存上,进程可以通过对这段内存的读取,从而实现对文件的读取,从而实现了内存共享。由于不同进程可直接读写内存,而无需任何数据的拷贝,共享内存的进程通信方式提高了系统效率。操作步骤如下所示:

- (1) 设置采集图像的各种属性(picture): ioctl(vd->fd, VIDIOCGPICT, &(vd->picture));
- (2) 初始化 video_mbuf, 获取所映射的 buffer 信息:

ioctl(vd->fd, VIDIOCGMBUF, &(vd->mbuf));

(3) 修改当前的 video_mmap 以及帧状态设置;

(4) 将 video_mbuf 与 mmap 绑定;

```
void * mmap (void * addr , size_t len , int prot ,
int flags , int fd , off_t offset );
```

(5) 内存映射中真正实现视频截取的 VIDIOCMCAPTURE ioctl(vd->fd, VIDIOCMCAPTURE, &(vd->mmap)), 开始一帧的截取,是非阻塞的,而是否截取完毕取决于 VIDIOCSYNC 的判断。

(6) 判断 VIDIOCSYNC,等待一帧截取结束:若成功,则一帧截取已完成,可开始下一次 VIDIOCMCAPTURE 视频获取;

(7) 视频采集工作结束后,调用 munmap 取消 video_mbuf 与 mmap 的绑定;

```
munmap(vd->map, vd->mbuf.size)
```

2) 通过直接打开的设备读取数据。

系统的图像格式、缓冲区大小等属性需由用户预先设置完毕。

用户预设置后,可调用 read, int read(待访问的文件描述符,指向待读写信息的指针,读写的字符个数)等相关函数。

通过上述操作,则摄像头采集到的数据已经保存到内存中。这些视频数据可以以文件的方式来存放,也可以经过一定的压缩算法压缩和网络传输到指定的数据中心或服务器。本设计采用的是第二种方法,其中压缩算法采用的是 MJPEG 压缩,生成视频数据流再进行网络传输。

2.3 视频的网络传输

网络传输模块主要是由客户端和网络服务器构成,用户通过客户端浏览器,输入 IP 地址即可实现对采集数据的实时播放。通过使用流媒体服务器 Spcaserv 来监控服务器,并负责网页传输、Java Applet 下载、控制对摄像头等操作。服务器的传输控制部分主要分成服务器端和客户端的通信过程,解析客户端所发送的信息,并调用相关的操作函数。

本系统使用 Socket 接口是 TCP/IP 网络的 API,流式 Socket (SOCK_STREAM) 是一种面向连接的 Socket,针对于面向连接的 TCP 服务应用^[10]。

Socket 编程步骤如下:

(1) 创建套接字。

```
int socket(int domain, int type, int protocol);
```

(2) 绑定套接字到 IP 地址和端口。

```
int bind(int sockfd, struct sockaddr * myaddr, int
addrlen);
```

(3) 客户端向服务器发送连接请求(connect)。

```
int connect( int sockfd, struct sockaddr_in * serv_
```

```
addr, int addrlen);
```

(4) 将套接字设为监听模式,准备接收客户请求。

```
int listen(int sockfd, int backlog);
```

(5) 等待客户请求的到来:当请求到来时,接受连接请求,并返回一个新的对应于此连接的套接字。

```
int accept(int sockfd, struct sockaddr_in * addr, int * ad-
drklen);
```

(6) 服务端用返回的套接字和客户端进行通信:

```
int send(int sockfd, const void * msg, int len, int
flags);
```

```
int recv(int sockfd, void * buf, int len, unsigned
int flags);
```

```
int sendto(int sockfd, const void * buf, int buflen,
unsigned int flags, const struct sockaddr_in * to, int to-
len);
```

```
int recvfrom(int sockfd, void * buf, int buflen, un-
signed int flags, struct sockaddr_in * from, int * from-
len);
```

(7) 结束传输。

```
close(socket);
```

```
int shutdown(int sockfd, int how);
```

视频服务器启动之后,以网络服务线程为主线程,并创建一图像采集线程,该线程不断地将采集的图像存入映射内存中。通过该操作,可实现多客户端同时连接到服务器,进行多线程操作。Socket 流程图如图 2 所示^[11]。

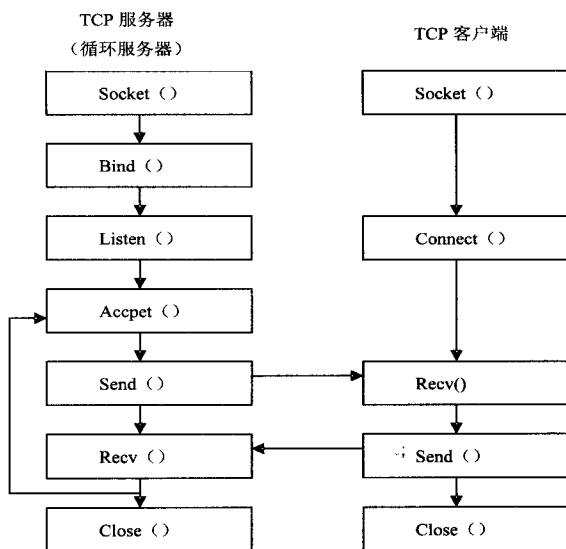


图 2 Socket 流程图

2.4 视频的实时显示

Java Applet,用 Java 语言编写的小应用程序,可直接嵌入到网页中,并能够加入一些特殊效果^[12]。本系统利用 Java Applet 完成视频的实时显示。其执行过程可概括如下:

(1)在网页中嵌入 Java 程序,并植入 Web 服务器定义的网页文档目录,启动服务器。

(2)在客户端 IE 中输入摄像头的 IP 地址,连接。

(3)成功连接后,Java Applet 程序包和网页被加载到客户端并执行。并发送视频服务器请求。

(4)服务器接收到请求后,把视频信号发送到 Java Applet 窗口并显示。

如果客户端主动断开连接,则客户端将不停地向服务器端发送请求,服务器端也就不断地发送视频信号,从而客户端也就得到了连续的视频图像。

3 结束语

文中提出了一种基于 Intel PXA270 嵌入式微处理器和嵌入式 Linux 网络摄像机的设计方案,系统采用先进的 MJPEG 编码标准,具有较高的压缩比,能够同时接收多个客户端的连接,加上 PXA270 嵌入式处理器的高效、低耗等特点,再加上网络的便捷,因此本系统具有性能稳定、结构简易、成本低廉等优点,在视频监控领域具有广阔的应用前景。

参考文献:

- [1] 杨福宝,王 建. 嵌入式系统及其在监控系统中应用研究[J]. 制造业自动化,2011(12):13-15.
- [2] Belbachir A N. Smart Cameras [R]. New York: Springer,

(上接第 167 页)

虚拟化技术来建设计算资源池,以解决资源利用率不足的问题^[13]。

3 结束语

“十二五”规划为云计算发展提供了机遇,云计算是整个 IT 产业发展的趋势,也是新一代互联网、物联网和移动互联的引擎和神经的中枢。通过分析典型的云计算方案,了解了各云计算方案的侧重点,对云计算有了深入的理解,将更有利于推动云计算的发展。

参考文献:

- [1] 工信部. 云计算是十二五规划中重点项目[EB/OL]. 2004-04-03. <http://www.cfi.net.cn/newspage.aspx?id=20110403000057&p=1>.
- [2] 《云计算》新版[EB/OL]. [2009-02-27]. <http://www.zenmeyang6.com/a/jisuan/jisuan/2009/1122/2107.html>.
- [3] 中国云计算网,什么是云计算?[EB/OL]. 2008-05-14. <http://www.cloudcomputingchina.cn/Article/ShowArticle.asp?ArticleID=1>.
- [4] De S K, Krishna P R. Clustering web transactions using rough approximation[J]. Fuzzy Sets and Systems, 2004, 148: 131-

2010.

- [3] 肖政宏,韩秋风,朱丽群. 基于 ARM 和 DSP 的远程视频监控系统的设计与实现[J]. 计算机工程与科学, 2006(9): 53-57.
- [4] 彭 涛,李声晋,芦 刚,等. 远程设备监控系统中嵌入式 Web 服务器的设计[J]. 机械与电子, 2008(1): 65-68.
- [5] 默罕莫德·默森,夏玮玮,沈连丰. 嵌入式视频监控服务器硬件的设计与实现[J]. 低压电器, 2004(12): 25-29.
- [6] Intel PXA27x Processor Family Developer's Manual [M]. [s. l.]: Intel Corporation, 2004.
- [7] 兰 婧,朱怡安,袁 磊. 基于 PXA270 嵌入式系统的 Bootloader 研究与实现[J]. 计算机工程与设计, 2009, 30(21): 4481-4483.
- [8] 曹少坤. 一种嵌入式网络摄像机的设计[J]. 微计算机应用, 2008, 29(10): 146-149.
- [9] 赵苍明,穆 煜. 嵌入式 Linux 应用开发教程[M]. 北京:人民邮电出版社, 2009.
- [10] Wehrle K, Pahlke F, Ritter H. The Linux Network Architecture Design and Implementation of Network Protocols in Linux Kernel [M]. Beijing: Tsinghua University Press, 2007.
- [11] Comer D E. Internetworking with TCP/IP: Principles, Protocol and Architecture [M]. 3rd ed. Beijing: Tsinghua University Press, 2003.
- [12] 陈 新,王建东,王成友,等. 基于 ARM 的高性能远程监控系统[J]. 微处理机, 2007(1): 86-89.

138.

- [5] 亚马逊 AWS[EB/OL]. [2011-05-27]. <http://aws.amazon.com/>.
- [6] Google App Engine[EB/OL]. [2010-04-21]. <http://wenku.baidu.com/view/c2e2db2f0066f5335a81213f.html>.
- [7] Ghorbani A A, Xu Xiaowen. A fuzzy markov model approach for predicting user navigation[C]//IEEE/WIC/ACM International Conference on Web Intelligence. [s. l.]: [s. n.], 2007: 307-311.
- [8] IBM 大中华区云计算中心[EB/OL]. [2010-05-25]. <http://wenku.baidu.com/view/89715d232f60ddccda38a025.html>.
- [9] 微软 Azure[EB/OL]. [2010-02-07]. <http://www.microsoft.com/windowsazure/>.
- [10] 浪潮发布国内首款云数据中心操作系统云海 OS[EB/OL]. [2011-05-31]. <http://www.jlpzj.com/viewthread.php?tid=212020>.
- [11] 华为云计算扬帆启航—云计算解决方案[EB/OL]. 2011-05-16. <http://www.enet.com.cn/solution/>.
- [12] 移动“大云”3.0 规划出炉[EB/OL]. 2010-12-20. <http://it.sohu.com/20101220/n278403396.shtml>.
- [13] 天云科技 Tcloud[EB/OL]. [2010-11-26]. <http://tcloud-computing.com.cn/index.php/company>.