

智能家居远程视频监控系统的设计与实现

付 蓉, 严建亮

(南京邮电大学 自动化学院, 江苏 南京 210003)

摘 要:针对传统模拟视频监控中图像质量差、无法联网、不可移植等不足,设计了一种基于嵌入式 Linux 的智能家居远程视频监控系统。远程视频监控系统以 ARM(Advanced RISC Machines)处理器 S3C2440 和 Linux 操作系统为核心平台。在内核中添加了各种所需的驱动程序,基于 V4L2 的应用程序对 USB 摄像头进行图像视频采集,采用 TCP/IP 网络协议,进行视频数据传输。主机端采用跨平台开发工具 Qt 开发应用程序,接收图像视频数据并显示图像视频。实验结果显示,能在远程端看到清晰流畅的视频流,系统运行稳定可靠。

关键词:视频监控;嵌入式;网络编程;智能家居

中图分类号:TP368.2

文献标识码:A

文章编号:1673-629X(2012)03-0137-04

Design and Implementation of Remote Video Monitor System at Smart Home

FU Rong, YAN Jian-liang

(Automation Institute, Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: Aimed at some shortages of the traditional simulation video monitor, such as bad quality image, unable to be online, unable to be transplanted and so on, design a smart home remoting video monitor system based on embedded Linux. The system uses embedded microprocessor ARM and Linux operating system as the core platform. Add all the necessary drivers in kernel. The application program of V4L2 captures image data by using a USB camera, and sends the image data based on TCP/IP. Qt cross-platform development tool is used by host computer to design an application to receive image data and display the image. The experimental results show that you can see the clear and smooth the video stream on computer. The system runs steadily.

Key words: video monitor; embedded; network programming; smart home

0 引 言

随着人民生活水平的提高和安全防范意识的增强,视频监控系统以其直观、准确、及时和信息内容丰富的优势,广泛应用于智能家居系统。视频监控经历了模拟监控、数字监控及网络监控三个主要阶段。传统模拟监控在图像质量、监控范围、联网上有较大的局限性,正逐步淘汰。

因此文中设计了一种基于嵌入式 Linux 的远程视频监控系统。该系统以 ARM 处理器 S3C2440 和 Linux 操作系统为核心平台,USB 摄像头采集视频数据,跨平台开发工具 Qt 开发应用程序。实验结果表明,该系统具有跨平台可移植性、能联网、稳定可靠、成本低廉等特点,具有广泛的应用价值。

1 系统总体结构及软硬件平台

嵌入式视频监控系统总体结构功能分为三个部分:远程端视频数据采集,PC 终端的视频显示,以及远程端和 PC 终端的视频数据通信。远程端 USB 摄像头负责视频图像数据采集,ARM 开发板上基于 V4L2 的应用程序采集视频图像数据到内存,用 socket 编程发送图像数据,服务端 PC 终端 Qt 应用程序接收视频数据并显示视频。这样就能实现远程视频监控了。视频监控总体结构[1~5]如图 1 所示。

(1) 硬件平台设计。开发板采用飞凌公司的 FL2440 型开发板,基于 S3C2440A 处理器^[6],主频 400MHz,64M 的 SDRAM,256M 的 NAND FLASH。要用到的主要配套外设:一个 USB HOST,一个 100M 网口(采用 DM9000AE)。USB 摄像头采用 VIMICRO USB PC Camera (ZC030X)。

(2) 软件平台设计。嵌入式 Linux 系统主要包括 bootloader、Linux 内核和文件系统三大部分。内核版本为 linux-2.6.30.9。个人 PC 机安装 Linux 操作系统:

收稿日期:2011-07-31;修回日期:2011-11-04

基金项目:中国博士后基金(20100481153)

作者简介:付 蓉(1974-),女,副教授,博士后,研究方向为复杂电力网络安全稳定控制;严建亮(1987-),男,福建莆田人,硕士,研究方向为复杂系统与网路控制。

ubuntu 8.10, Linux 操作系统安装交叉编译器 arm-linux-gcc-4.3.2、Qt 的集成开发环境 QtCreator 等。

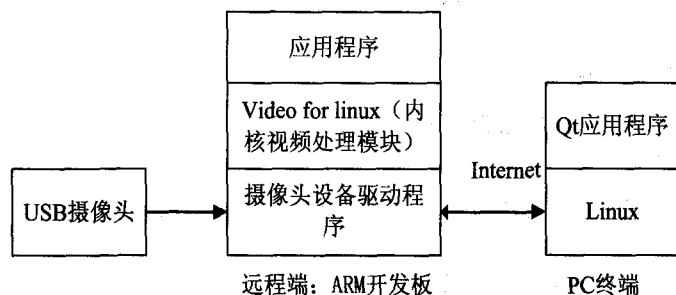


图 1 视频监控总结构图

2 基于 V4L2 的视频采集

2.1 V4L2 简介

Linux 下的图像应用程序设计一般都是基于 video for linux 开发, video for linux 简称 V4L, 现在已经发展到 V4L2。V4L 是 Linux 操作系统下用于采集图片、视频和音频数据的 API 接口, 配合适当的视频采集设备和相应的驱动程序, 可以实现图片、视频、音频等的采集。在远程会议、可视电话、视频监控系统和嵌入式多媒体终端中都有广泛的应用。

V4L2 不仅为驱动程序的编写提供统一的接口, 也方便应用程序的编写和移植。它屏蔽了不同摄像头设备的差异, 应用程序不需要修改, 就像 QQ 视频, 大部分摄像头都能用就是这个道理。

2.2 USB 摄像头驱动的安装

USB 摄像头驱动在 Linux 内核配置。对于 Linux 内核, 如果支持视频类设备, 需要在终端模式下输入 make menuconfig, 在 Linux 内核配置界面中, 配置 video for linux 和 video capture adapters 项, 并进入到 video capture adapters 选择所使用的摄像头驱动。

2.3 基于 V4L2 的 USB 摄像头的图像具体采集流程

基于 V4L2 利用 USB 摄像头进行图像采集, 主要的工作就是调用 V4L2 给应用程序提供的接口函数, 通过 V4L2 提供的接口函数来调用 USB 摄像头驱动, 完成图像和视频的采集功能。基于 V4L2 的 USB 摄像头的图像具体采集流程^[7]如图 2 所示。

视频图像采集应用程序主要是通过 ioctl 函数和相关的命令字来调用 V4L2 的:

```
int ioctl(int fd, int request, void * argp);
```

三个参数分别表示设备描述符、控制命令字和控制命令参数。ioctl 主要的命令字和对应功能如表 1^[8]所示。

2.4 基于 V4L2 的 USB 摄像头的图像采集过程

应用程序申请若干个视频数据的帧缓冲区, 这些帧缓冲区在内核空间。应用程序再通过内存映射方法

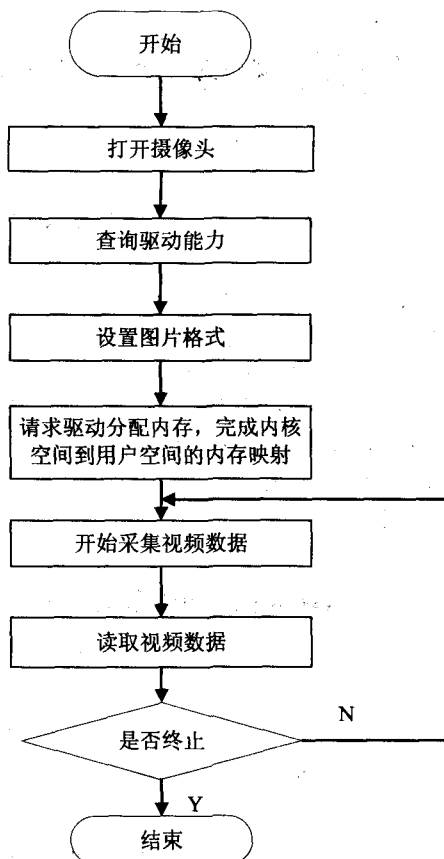


图 2 基于 V4L2 的 USB 摄像头的图像具体采集流程

表 1 ioctl 主要的命令字和对应功能

命令字	功能
VIDIOC_ENUM_FMT	获取当前驱动支持的视频格式
VIDIOC_QUERYCAP	查询驱动的修剪能力
VIDIOC_REQBUFS	分配内存
VIDIOC_QUERYBU	把 VIDIOC_REQBUFS 中分配的数据缓存转换成物理地址
VIDIOC_QBUF	把数据从缓存中读取出来
VIDIOC_STREAMON	开始视频显示函数
VIDIOC_DQBUF	把数据放回缓存队列
VIDIOC_STREAMOFF	结束视频显示函数

(mmap), 将申请到的内核空间帧缓冲区的地址映射到用户空间地址, 这样就可以直接处理帧缓冲区的数据。在驱动程序处理视频的过程中, 定义了两个队列: 视频采集输入队列和视频采集输出队列, 前者是等待驱动存放视频数据的队列, 后者是驱动程序已经放入了视频数据的队列。

应用程序需要将上述帧缓冲区在视频采集输入队列排队, 然后可启动视频采集。启动视频采集后, 驱动程序开始采集一帧数据, 把采集的数据放入视频采集输入队列的第一个帧缓冲区, 一帧数据采集完成, 也就是第一个帧缓冲区存满一帧数据后, 驱动程序将该帧缓冲区移至视频采集输出队列, 等待应用程序从输出队列取出。驱动程序接下来采集下一帧数据, 放入第二个帧缓冲区, 同样帧缓冲区存满下一帧数据后, 被放入视频采集输出队列, 如图 3 所示:

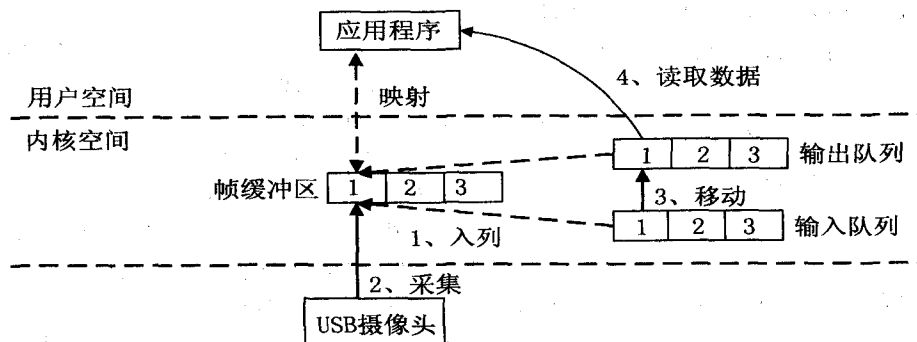


图3 图像采集过程

应用程序从视频采集输出队列中取出含有视频数据的帧缓冲区,处理帧缓冲区中的视频数据,如存储或压缩。

最后,应用程序将处理完数据的帧缓冲区重新放入视频采集输入队列,这样可以循环采集。

3 视频发送与接收

3.1 Linux 网络应用程序设计模型

Linux 下网络应用程序的开发一般都基于 socket (套接字) 来开发,socket 层位于 TCP/UDP 层之上,屏蔽了 TCP/IP 协议的细节,使得应用程序访问网络接口就像读写文件一样简单。通过 socket,大大简化了网络应用程序的设计。

3.2 面向连接的套接字应用程序流程图

这里使用面向连接的 TCP 传输控制协议。

面向连接的套接字工作过程^[9]如下:服务器首先

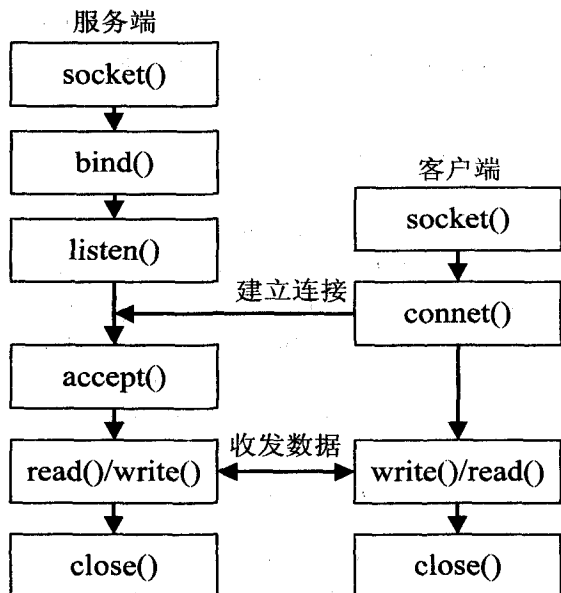


图4 面向连接的套接字应用程序流程图

启动,通过调用 socket 函数建立一个套接字,然后调用 bind 将该套接字和本地网络地址联系在一起,再调用 listen 使套接字做好侦听的准备,并规定它的请求队列的长度。客户端在建立套接字后就可以调用 connect

和服务端建立连接。服务端调用 accept 返回一个新的套接字来接收连接。这样,客户端和服务端就可以通过调用 read 和 write 来发送和接收数据。最后,待数据传送结束后,双方调用 close 关闭套接字。面向连接的套接字应用程序流程

图^[10,11]如图4所示。

所以,基于 V4L2 的应用程序拍完图像数据,直接用 socket 客户端发送数据:

```
write ( sockfd, buffers [ buf. index ]. start, buffers
[ buf. index ]. length )
```

3.3 QTcpServer 类接收视频图像数据

基于 V4L2 的应用程序拍完图像数据,并直接用 socket 客户端发送数据,PC 终端上面的应用程序接收图像数据并显示视频。这里的应用程序用 Qt 集成开发环境 QtCreator 开发,接收数据采用 Qt 里面的 QTcpServer 类等实现,代替 socket 服务端。

3.3.1 Qt 简介

Qt 是一个完整的 C++ 应用程序开发框架,它包含一个类库和用于跨平台开发及国际化的工具。Qt 集成开发环境 QtCreator 类似 Windows 环境下的 VC++。

信号和槽机制是 Qt 的一个主要特征,在 Qt 中采用信号和槽来实现对象部件之间的通信。这样的话,Qt 里面就很容易实现多线程编程。

3.3.2 QTcpServer 接收视频图像数据

Qt 中用 QTcpSocket、QTcpServer 类^[12]实现网络通信,它们是在 socket 的基础上封装各种类,原理是基于 socket 的。QTcpServer 是服务端类。QTcpSocket 是 TCP 套接字类。基本流程如下:

1) 侦听 QTcpServer. listen (QHostAddress:: Any, 5023), 5023 是端口号。

2) 有连接请求时,QTcpServer 产生 newConnection () 信号,调用 nextPendingConnection () 接受请求并返回 QTcpSocket,就可以用这个类进行通信了。

3) 每收到一个数据包,QTcpSocket 产生 readyRead () 信号,调用 QTcpSocket. readAll () 读取数据,存到 QByteArray bytearray 类型数组里。

4) 获取完整的一帧数据。QTcpSocket. readAll () 读取的数据并不是一帧数据,而是一帧数据的各个分段部分,需要读到一个完整的帧大小才能显示。

4 图像视频显示

图像视频显示采用 QPixmap 类和 QLabel 类。先

调用 QPixmap. loadFromDate (bytearray2) 函数获取数据,再加载到 QLabel 显示, QLabel. setPixmap (QPixmap)。

当 ARM 远程端不断采集图像数据,PC 终端不断接收并显示图像数据,就能在 PC 终端形成视频流。

5 系统结果与展望

通过超级终端下载采集应用程序到开发板 Linux 操作系统里。通过 ping 命令查看网络连接是否正常。

打开 PC 终端 Qt 应用程序,点击 Start 按钮,服务端处于侦听状态,启动 ARM 板上应用程序,就能在 PC 终端看到清晰稳定的视频流。

本项目的下一步工作是研究图像编码和解码,提高帧传输速率,以提高清晰度。加入报警模块。ARM 处理器对图像进行处理,定位出人脸区域,提取出人脸特征,并将其与数据库中用户的人脸特征进行对比^[13],如果不符合,则立即通过 Internet 向用户发送报警信息。

6 结束语

文中详细分析了视频监控系统的实现步骤,从基于 V4L2 的应用程序图像采集,到使用 socket 发送图像数据,以及最后的 Qt 的接收图像数据并显示图像,从而实现了整个视频监控系统。详细分析了基于 V4L2 的 USB 摄像头的图像具体采集过程和采集流程、面向连接的 socket 编程基本流程和 QTcpSocket 类。

嵌入式视频监控在安防系统中具有巨大的应用价值。

参考文献:

- [1] 唐人财,刘连浩. 基于嵌入式 Linux 远程图像监控系统的设计[J]. 计算机与现代化,2010(11):31-38.
- [2] 伍 俭,罗桂娥. 基于 B/S 模式的嵌入式视频监控系统的的设计[J]. 微型机与应用,2010(4):74-76.
- [3] 杨 念,李 峰. 基于 B/S 的嵌入式视频监控系统的的设计与实现[J]. 计算机工程与设计,2008,29(21):5576-5579.
- [4] 李华毅. 基于嵌入式技术的智能家居监控系统的设计与实现[D]. 武汉:武汉理工大学,2008.
- [5] 焦 焰,易小波,李仁发. 基于嵌入式 Internet 的远程视频监控系统设计[J]. 计算机技术与发展,2009,19(5):176-179.
- [6] Ji Wangkang, Yang Jia, Hong Yongqiang. BSP development of WinCE system for vehicle navigation device based on S3C2440 [C]//8th International Conference on Electronic Measurement and Instruments. [s. l.]: [s. n.], 2007:389-391.
- [7] Dirks B, Schimek M H, Verkuil H, et al. Video for Linux Two API Specification (Revision 0.24) [S]. 2008.
- [8] 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.
- [9] 刘学勇,陈建伟. 精通 Linux C 编程[M]. 北京:清华大学出版社,2009.
- [10] 华清远见嵌入式培训中心. 嵌入式 Linux 应用程序开发标准教程[M]. 第 2 版. 北京:人民邮电出版社,2009.
- [11] 彭铁钢,刘国繁,曹少坤,等. 基于 ARM 的嵌入式视频监控系统设计[J]. 计算机工程与设计,2010,31(6):1191-1193.
- [12] Nokia Corporation. Qt 4.6: All Qt Classes [CP/DK]. 2010.
- [13] 刘 成,鲍可进. 基于图像处理的嵌入式自动报警系统设计[J]. 计算机工程与设计,2007,28(17):4198-4199.

(上接第 136 页)

极的影响。

参考文献:

- [1] 陆汝铃. 人工智能(上)[M]. 北京:科学出版社,1995.
- [2] Rich E. Artificial intelligence [M]. [s. l.]: McGraw-Hill, 1983:340-341.
- [3] Illingworth V. Dictionary of Computing [M]. 4th ed. Oxford: Oxford University Press, 1996.
- [4] 中国大百科全书·心理学[M]. 北京:中国大百科全书出版社,1991.
- [5] 王同忆. 新世纪现代汉语词典[M]. 北京:京华出版社,2001.
- [6] Reed D. A balanced introduction to computer science [M]. 3rd ed. [s. l.]: Prentice Hall, 2011.
- [7] Lapedes D N. McGraw-Hill Dictionary of Physics and Mathematics [M]. New York: McGraw-Hill Inc, 1978:686-689.
- [8] 洪 龙,陈燕俐,朱梧楦. 80486 极小指令集及其构造性证明[J]. 南京邮电大学学报,2006,26(4):57-64.
- [9] Tanenbaum A S. Structured Computer Organization [M]. 4th ed. [s. l.]: Prentice Hall, 1999.
- [10] 朱梧楦,肖奚安. 数理逻辑导引[M]. 南京:南京大学出版社,1995:63-78.
- [11] Bouzy B, Cazenave T. Computer go: an AI oriented survey [J]. Artificial Intelligence, 2001, 132(1):39-103.
- [12] Narayanan A. On being a machine (vol 2): philosophy of artificial intelligence [M]. [s. l.]: Ellis-Horwood Ltd., 1990.