

基于 ARM/WiFi/QT 的无线视频监控系统的开发^①

吴新生

(广东科学技术职业学院 广州学院, 广州 510640)

摘 要: 利用 ARM 和 QT 开发了一款基于 WiFi 网络的无线视频监控系统. 系统由远程视频采集端、WiFi 网络和监控中心三部分构成. 远程视频采集端采用 PXA270 ARM10 微处理器为核心的硬件平台, 并在其上建立 Linux 操作系统, 完成视频的采集、H.264 编码压缩和传送. 监控中心使用 QT 利用多线程技术开发了客户应用软件, 实现监控视频的解码和显示. WiFi 网络实现视频采集端和监控中心间的视频传输. 使用效果表明, 该系统能稳定实时地显示一路或多路视频监控图像, 很好地达到了设计的要求, 在各领域具有广泛的应用前景.

关键词: 无线视频监控; ARM; WiFi; QT; H.264

Development of Wireless Video Monitoring System Based on ARM/WiFi/QT

WU Xin-Sheng

(Guangzhou College, Guangdong Institute of Science and Technology, Guangzhou 510640, China)

Abstract: A wireless video monitoring system was developed with ARM and QT based on WiFi network. The system was composed of the remote embedded video acquisition terminal, WiFi network and the monitoring center. In this system, the hardware platform with PXA270 ARM microprocessor chip as its core is used to capture video and the Linux system is built up on it. The client software is developed by QT and the multi-thread technology as well in order to display the video from acquisition terminal by WiFi network. The experiment results showed that the real-time display of one-way or multi-way video monitoring image could be achieved. The system could satisfy the requirements of design and would be widely used in various fields.

Key words: Wireless Video Monitoring; ARM; WiFi; QT; H.264

视频监控技术已经在教育、旅游、交通、商业、水利和楼宇等领域得到了广泛的应用. 视频监控技术经历了模拟视频监控、数控视频监控和数字网络视频监控三个阶段^[1]. 数字网络视频监控系统是现代视频监控技术使用最广泛的系统. 网络视频监控系统主要采用有线传输和无线传输两种方式. 由于有线网络传输方式存在布线复杂、网络结构和监测点固定等问题, 很难适应监控的环境变化大的要求, 而无线网络视频监控系统能很好地解决这些问题. 虽然也有人使用 3G 等无线网络实现了无线视频监控系统^[2], 但是由于网络本身的通信速度和使用成本限制了它的使用范围.

由于嵌入式技术的发展, 嵌入式微处理器具有可靠高、成本低, 体积小和实时性高等特点, 将其应用到

视频监控系统的开发将大大降低系统的成本, 提高系统的稳定性. 本文将采用高性能的嵌入式 ARM 处理器, 结合嵌入式 Linux 操作系统, 将 USB 摄像头获取的视频经 H.264 编码压缩后通过无线 WiFi 网络传输到监控中心, 监控中心通过 QT 开发的客户端软件, 实现视频的接收、解码和显示.

1 无线视频监控系统的总体设计

无线视频监控系统主要包括远程视频采集端、WiFi 网络和监控中心三大部分. 远程视频采集端通过 USB 接口的摄像头采集到的视频数据, 传输到 PXA270 ARM 中, 并在 ARM 上进行 H.264 压缩编码后, 通过无线 WIFI 网络传输到监控中心, 在监控中心

^① 基金项目: 广东省自然科学基金(8451064007000003)

收稿时间: 2013-01-18; 收到修改稿时间: 2013-02-26

进行实时的解码、显示和贮存,以供管理人员进行分析和报警.具体结构图参见图 1.

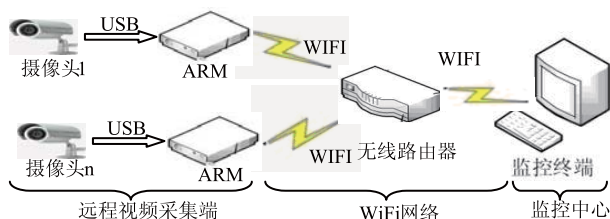


图 1 系统总体结构图

2 远程视频采集端的开发

该模块主要完成监控视频的采集和压缩后,并通过 WiFi 网络把压缩后的视频传送到监控终端.系统采用以 PXA270 ARM10 微处理器芯片为核心的硬件平台,操作系统选用 Linux 操作系统,视频采集使用了 USB 接口摄像头. ARM 通过 UART 接口和 WiFi 模块相连. WiFi 模块使用 Marvell 的 88W8686 模块,该模块兼容 802.11b/g 协议,接收的范围变得更大,保证了视频数据的传输速率在 25 帧/秒以上. 视频采集端的具体硬件结构图如图 2 所示.

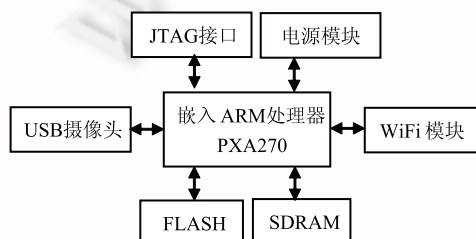


图 2 视频采集硬件结构图

在视频采集端,先建立了 Linux 操作系统,并移植了 gspcav 开源摄像头的驱动程序,并使用 Linux 中的 V4L API 实现了视频的采集,采集到的视频经 H.264 编码压缩后,通过 WiFi 网络使用基于 TCP/IP 协议的 socket 发送视频数据到监控中心.

2.1 嵌入式 Linux 操作系统的构建

主要包括 Bootloader Blob 的移植、Linux 2.6 内核的移植和根文件系统的制作三个方面.

2.1.1 Blob 的定制和移植^[3]

Blob 的启动过程分为两个阶段:Stage1 和 Stage2. 其中 Stage1 主要为依赖于处理器体系结构和板级硬件初始化的代码,采用汇编语言实现,这一阶段代码的起

点是 start.S; 而 Stage2 主要采用 C 语言实现,这不仅能实现更复杂的功能,而且代码会具有更好的可读性和可移植性,这一阶段代码的起点是 main.c. 把 Blob 移植到 PXA270 硬件平台上,主要进行以下两方面的工作:

① 根据平台修改或者重新编写硬件初始化代码.

② 添加 PXA270 及其硬件平台构架相关的头文件,在这个文件中指定 Blob、内核、Ramdisk 等在 Flash 的地址及其被加载到 RAM 中的地址, CPU 相关的宏定义,硬件平台相关的宏定义等等.

2.1.2 Linux 内核的定制、编译和下载^[4]

① 进入 Linux2.6 内核所在的目录,修改相应 Makefile 的变量 ARCH 为 arm 及 CROSS_COMPILE 的值为所使用的交叉编译器的前缀.

② 执行 make menuconfig 命令,通过定制菜单对 Linux 的功能进行选择和配置.

③ 输入 make zImage 命令,进行内核编译,并生成压缩的内核影像文件 zImage.

④ 使用 Blob 提供的网络功能把 zImage 下载到 SDRAM 并烧写到 Flash 中.

2.1.3 根文件系统的构建

使用 BusyBox 构建根文件系统,主要包括以下步骤:

① 进行 busybox 所在的目录,执行 make menuconfig 命令设定所使用的交叉编译器的前缀和安装的目录,并选择所要的各个命令模块.

② 使用 make 命令进行编译.

③ 使用 make install 命令把编译好的命令安装到指定的安装目录中.

④ 把编译好的有关标准库和 qte 库文件拷贝到 lib 目录中.

⑤ 有关 Linux 系统中的 inittab 等有关启动文件的编辑,以完成系统启动的库文件和环境参数的设定.

2.2 视频采集的实现、编码和传输

2.2.1 开源 USB 摄像头视频驱动的移植

gspcav 是一款在 Linux 操作系统上使用的万能摄像头驱动程序,大约能支持 100 多种摄像头.它属于开源软件,源程序可以在网络上进行下载.下载后,解压并进行编译,得到 gspcav.ko 的驱动模块,使用 insmod 命令进行加载摄像头模块驱动.

2.2.2 视频采集程序的实现

video4linux 是 Linux 操作系统用于获取视频和音

频数据的 API 接口, 现有的 video4linux 有 v4l 和 v4l2 两个版本, 由于移植的开源摄像头驱动仅支持 v4l 接口, 所以主要用到 v4l API 进行编程. 利用 v4l API 获取视频图像一般有以下几步:

- ① 打开视频设备;
- ② 设置设备的属性;
- ③ 获得采集到的图像数据;
- ④ 处理采集到的图像数据;
- ⑤ 关闭设备

2.2.3 视频数据的 H.264 编码压缩

为了提高网络的传输效率, 需要将采集好的视频数据进行 H.264 编码压缩. 为了节省系统的成本, 使用开源软件 x264 来进行 H.264 软编码^[5]. 在编译前, 首先把软件中不需要的解码和传输部分进行了删除, 并对部分的代码进行了改写优化, 并将编码中复杂的程序使用 ARM 汇编程序进行改写, 并专门对于 PXA270 的特点进行了代码优化. 修改和优化后的 x264 软件通过交叉编译后移植到 PXA270 ARM 平台上, 优化后的 H.264 编码器每秒能处理 25 帧以上的 QCIF 图像, 满足了监控视频高实时性的要求.

2.2.4 视频数据的网络传输

使用 QT 中的 QTcpSocket 类^[6], 实现了 TcpSocket 服务端, 进行视频数据的传输. 具体的流程参见图 3.

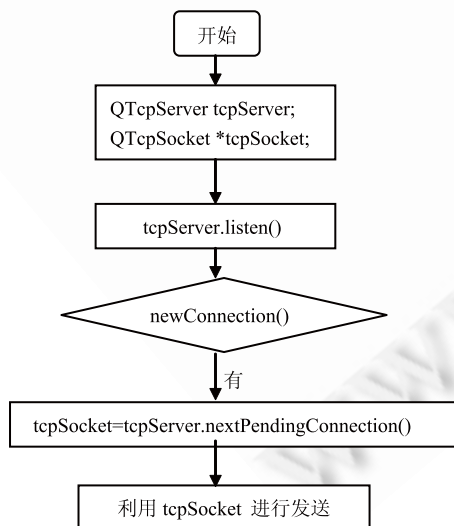


图 3 QTCPsocket 服务端的流程图

3 基于QT的视频客户端软件的开发

该软件运行在监控中心的监控终端上, 主要完成以下三个功能: ①接收来自视频采集端的视频数据;

②进行 H.264 解码接收到的视频数据; ③显示解码后的视频图像. 采用 QT 开发以上功能. 具体实现中, 为了提高实现视频显示的效率, 采用了多线程技术, 满足实时显示视频画面的要求. 主线程在主窗口绘图, 实时显示监控视频; 子线程建立 TCP 连接, 接收来自远程视频采集端的数据, 当接收完一帧的数据进行 H.264 解码后, 再发射信号给主线程, 让主线程来绘图显示.

3.1 视频数据接收程序的开发

因为视频采集端使用 TCPSocket 协议来传输图像数据的, 接收程序要使用 TCPSocket 协议来建立网络连接和数据传输. 使用了 QT 中的 QTcpSocket 类可实现 TCPSocket 客户端接收程序, 具体流程参见图 4. 这部分功能是放在子线程中, 具体就是使用 QT 提供的线程类 Qthread 来达到这个功能^[6]. 当调用 QThread 类的 start()函数时, 会自动运行子线程的 run()函数. 视频接收的程序功能都在 run()函数中实现.

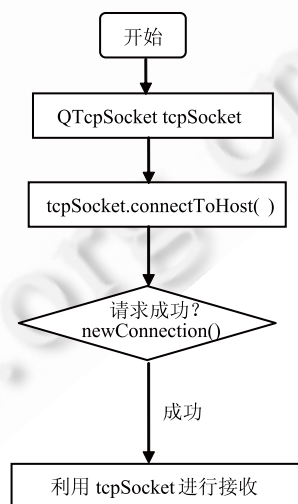


图 4 QTCPsocket 客户端的流程图

3.2 视频数据的 H.264 解码程序

具体使用了开源软件 x264 中的解压部分中的函数来实现解码. 当接收到一帧数据后, 调用解码函数来实现. 具体功能代码也是在子线程实现的, 具体调用代码在 run()函数中. 完成解码后, 发送 senddata 信号告诉主线程完成数据的显示.

3.3 视频图像的显示程序

使用 QT 中的 QPainter 类实现, 使用公有函数

(下转第 62 页)

从图中任一活动节点进行回退操作,流程的节点状态及数据也跟着回到之前,正确实现了流程回退。如图中运营部审核节点进行回退至新建申请目标节点,此时处于 and 分支的项目经理审核节点将被关闭,目标节点与当前节点的状态的改变都被记录下来,回退成功,重新执行新建申请节点,运营部审核、项目经理审核节点都被重新执行,这也证明了之前流程的回退时正确的。其余活动节点的回退操作也都实现了正确回退。

4 结语

本文针对现有工作流对用户自定义支持的不足,基于 flex 技术和 jbpm 开源工作流设计并实现了一套自定义工作流模型。一方面,利用 flex gef 框架,flex 技术为用户定义工作流的图形化设计和编辑提供了良好的体验,gef 框架实现了程序的松耦合,提高了灵活性及可扩展性;另一方面,通过动态路由解决了流程多跨步回退和分支回退的问题。此系统目前应用于某电信运营商的运营门户中,高效的实现业务工作流程的运行。今后的工作将在现有的系统模型上加以改进,进

一步降低耦合,完善系统架构。

参考文献

- 1 兰天,曲鹏东,孙高飞,等.Flex 企业应用开发实践.北京:机械工业出版社,2010.48-52.
- 2 刘晓菲,宋朝晖.浅谈基于 Flex 技术的 RIA 设计.电脑知识与技术,2010(11):125-128.
- 3 江虹莹.基于 JBPM 审批流程引擎的研究与实现[学位论文].成都:电子科技大学,2008.
- 4 王瑞玲.RIA 及其基于 Flex 的应用开发模型.科技情报开发与经济,2007,17(14):206-207.
- 5 WfMC. Workflow Management Coalition Technology and Glossary (WfMC-TC-1011). Technical Report Workflow Management Coalition,Brussels,1996.
- 6 彭晓川.基于 Flex 的 RIA 与 J2EE 应用的整合.电脑与电信,2008,15(2):37-39.
- 7 张伟滨.自定义工作流的功能分析.信息科技,2007,2(2):149-150.
- 8 刘利坤.基于 JBPM 和轻量级 J2EE 的办公自动化系统的研究与实现[学位论文].长春:东北师范大学,2009.

(上接第 78 页)

begin(), drawImage(), end()函数来实现绘图事件。主线程通过接受子线程的 senddata 信号来绘制图形以及调用 update()函数来重绘,从而来实现监控视频的实时显示。

4 结语

本文采用嵌入式方法开发的无线视频监控系统,已在学校的实验室监控系统中使用,视频的传输速度在 11Mbit/s-54Mbit/s 之间,达到了实时传输视频的需求,取得了不错的使用效果。该系统由于开发的设备成本低,性能稳定,并解决了传统的有线视频监控系统的问題,将具有广泛的应用前景。

参考文献

- 1 朱长坡.浅析网络监控应用发展.数字技术与应用,2010(6):172.
- 2 李波,卢文科,朱文超.基于 3G 和 H.264 的无线视频监控系统的設計.微计算机信息,2011,27(5):78-80.
- 3 李建国,罗代升,吴小强.Blob 在 Intel PXA270 上的设计与移植.成都信息工程学院学报,2008,23(2):132-136.
- 4 潘学文,文汉云.基于 PXA270 的嵌入式 Linux 系统移植构建.微处理机,2011,33(5):89-91.
- 5 孔群娥,曾学文,刘马飞.基于 x264 实现 H.264 的时域可伸缩编码.微计算机应用,2011,32(5):24-28.
- 6 丁林松,黄丽琴.Qt 4 图形设计与嵌入式开发.北京:人民邮电出版社,2003.