

基于 Linux 的嵌入式网络视频监控系统研究与设计

Research and Design of Embedded Network Video Monitoring System Based on Linux

郝卫东 李 静 (桂林电子科技大学 机电工程学院 广西桂林 541004)

摘 要: 网络视频监控已经在嵌入式领域得到了发展,但目前视频监控硬件设施比较昂贵,不能普遍使用。本文提出了一种以 ARM9 处理器为开发硬件平台和嵌入式 Linux 系统为软件开发环境的新方法,采用中星微 zc301 摄像头作为视频前端采集,利用 TCP/IP 协议技术实现网络通信。系统内部嵌入移植的 Web 服务器,用户可以通过浏览器进行远端访问。该系统实现了局域网的环境监控,开发成本低,使用方便而且性能高。

关键词: 嵌入式 视频采集 Web 服务器 视频监控 浏览器

1 引言

随着嵌入式处理器和开源 Linux 系统的广泛应用,网络视频监控向着嵌入式方向发展已经成为一种趋势。目前已经在该领域得到了初步发展,但是由于前端监控系统的硬件设施昂贵,成本较高,不能普遍使用。低廉的设备又不能满足用户的动态可视化以及稳定性和可靠性的要求。针对用户的需求,本论文设计了低成本高效能的方案,运用当前的网络技术、视频压缩技术和嵌入式技术等设计基于嵌入式 Linux 系统的网络视频监控系统。该系统以嵌入式 Linux 系统和嵌入式控制器 S3C2410 为核心平台,在这个平台上建立 web 服务器和视频服务器。利用 TCP/IP 协议技术实现网络通信,把 USB 数字摄像头采集到的图像进行 JPEG 压缩,然后通过网络传输到远计算机客户端。客户端方采用 Java Applet 技术实现图像的接收并完成显示功能。客户端的用户可以通过具有 java 插件的浏览器实现监控。实验证明该系统的稳定性、可靠性较高。

2 系统整体模块设计

本系统主要由嵌入式系统和网络通信两大系统组成。嵌入式系统由嵌入式处理器 S3C2410 和 Linux 系统组成,网络系统由 Web 服务器和客户端以及实现通信的通道组成。系统的设计采用模块化设计,本系统由嵌入式系统模块、视频采集模块、网络通信模块和客

户端组成。各个模块之间的关系如图 1 所示。



图 1 系统整体结构框图

嵌入式处理器 S3C2410 是一款基于 ARM920T 内核的 16/32 位 RISC 微处理器。内部带有全性能的 MMU(内存处理单元),它适用于设计移动手持设备类产品,具有高性能、低功耗、接口丰富和体积小等优良特性^[1]。该处理器集成了 64M SDRAM、64M Nand Flash、1M Boot Flash、RJ-45 网卡、音频输入与输出、USB Host、USB slave、标准串口、SD 卡插座等。用户可以将代码存放在 NAND Flash 中,并从 NAND Flash 启动。内置的 NAND Flash 将访问控制端口,并将引导代码(vivi)自动加载到内部 SRAM 运行,之后 SRAM 中的引导代码将操作系统内核映像(zImage)以及根文件系统加载到 SRAM 并运行^[2]。

3 系统软件的设计实现

3.1 视频采集模块实现

Linux 内核驱动里面已经支持了诸如 ov511 芯片的

摄像头,但是这些摄像头采集的效果不好。本系统采用中星微 zc0301 芯片的摄像头,该摄像头采用 cmos 传感器,价格低廉,性能高,usb 接口协议灵活,同时具有同步和异步数据传输,兼容性好,而且提供了良好的图像处理及对外接口功能。目前 linux 驱动里面不支持该类芯片的驱动。为了使摄像头能够在系统中工作,需要加载该驱动。驱动在下载地址 <http://mx-haard.free.fr/spca50x/embedded/KernelPatch/>。系统 linux 内核版本使用的是 linux2.6.11,需要下载 2.6 版本的内核补丁。然后对内核进行打补丁,这样内核就支持该芯片的驱动。

在 USB 摄像头被驱动后,只需要再编写视频采集的应用程序就可以了,视频图像的采集是通过 Video4Linux 实现的。视频采集程序的流程如图 2 所示。

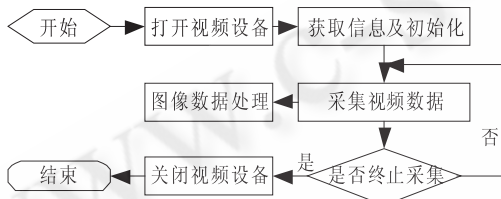


图 2 视频采集程序的流程

Video4Linux 是 Linux 中关于视频设备的内核驱动,它为针对视频设备的应用程序编程提供一系列接口函数。在视频采集中,应用程序首先通过 open() 函数打开视频采集设备并返回一个文件描述符号,这个文件描述符就代表了捕获的设备硬件。成功开启设备档后,调用 ioctl() 取的设备文件的相关信息,并且将取的信息放到 video capability 结构中,同样调用 ioctl() 将视频窗口信息放到 video picture 结构中。视频设备的关闭通过 close() 函数实现的。通过这些函数,可容易打开、读写、关闭等基本操作。

3.2 网络通信模块实现

网络通信模块负责将摄像头采集的视频图像数据通过网络传输到远端计算机。监控系统在实现方式上采用了“浏览器-服务器”结构,即在视频采集端建立嵌入式 Web 服务器,在客户端的 Web 浏览器上输入 IP 地址就能接收到从服务器端传过来的视频信息,它包括嵌入式 Web 服务器设计和视频应用程序的开发。

3.2.1 嵌入式 Web 服务器设计

Web 服务器和客户端的浏览器是通过超文本传

送协议 HTTP 进行通信的^[3]。其工作原理大致可分为三个步骤,第一步,Web 浏览器向一个特定的服务器发出 Web 页面请求,第二步,Web 服务器接收到 Web 页面请求后,寻找所请求的 Web 页面,并将所请求的 Web 页面传送给 Web 浏览器,第三步,Web 服务器接收到所请求的 Web 页面,并将它显示出来。Web 服务器工作原理如图 3 所示。

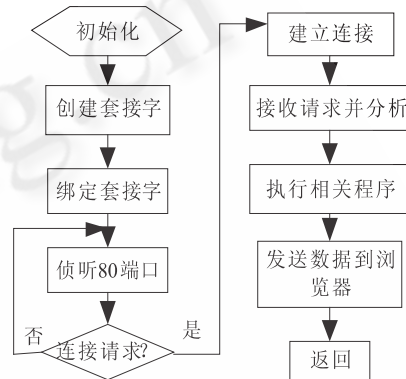


图 3 Web 服务器工作流程图

嵌入式 Web 服务器是移植 Web 服务器到嵌入式领域的一种服务器。它仍然是以 TCP/IP 为底层的通信协议^[4]。嵌入式 Web 服务器简化了系统结构,将信息采集和信息发布都继承到嵌入式设备中,具有标准的接口形式和通信协议,内嵌的 Web 服务器同样可以基于浏览器方式的操作和控制界面。

目前嵌入式领域中最典型的 Web 服务器有 boa 和 thttpd。Boa 是一个非常小巧的 Web 服务器,可执行代码只有约 60KB,占用系统的资源少,而且速度快和安全性能好^[5]。本系统采用的就是 boa 服务器,就是把 Boa 服务器的移植到本系统中。首先从 www.boa.org 下载 boa 源代码,目前最新版本 boa-0.94.13。解压到 linux 系统目录下,修改 Makefile。只需要把 CC = gcc 改成 CC = arm-linux-gcc 和 CPP = gcc - E 改成 CPP = arm-linux-gcc -E 就可以了。然后进行编译,把生成的 boa 应用程序下载到开发板上 linux 系统中的/bin 目录下,让 boa 在系统启动时候自动运行,这样在开发板上就移植了 boa 服务器。

3.2.2 网络视频应用开发

视频信息通过摄像头采集到的图像需要通过网络传送另一端通过浏览器显示出来,因而网络视频应用的开发包含视频服务器端程序设计和客户端程序

设计。

(1) 视频服务器端程序设计

视频服务器端工作主要是与客户端建立连接,并根据客户端用户的指令,调用相应的函数处理客户端的控制要求。服务器端程序由 C 语言开发。主程序的设计首先初始化监控设备的数据结构,包括采集模式、显示模式和端口初始化设置等,初始化并分配内存并以可读写的方式打开视频设备。然后系统创建了一个 grab 线程,grab 函数通过调用 v4lGrab() 采集图像并传输到缓冲区。然后创建一个 Socket 对象并监听,然后用 Accept 方法监听是否有客户端的连接请求。如果没有客户端连接则进程阻塞,继续监听,当有客户端连接请求则创建一个新的 Socket 对象,然后启动视频服务线程 service 来与客户端通信。视频服务器工作流程如图 4 所示。

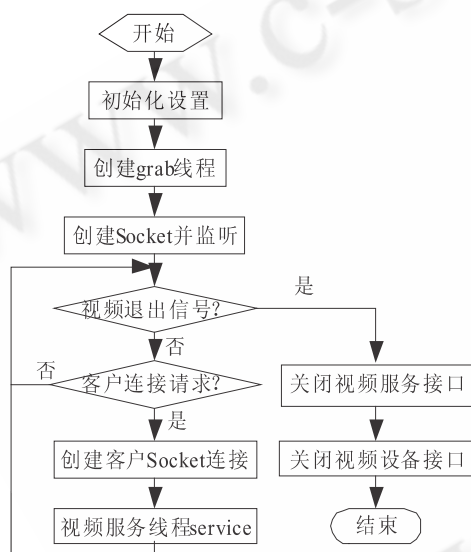


图 4 视频服务器工作流程

(2) 客户端程序设计

在本系统中,远程的 Web 用户查看视频采集的图像信息是通过 Applet 与视频服务器进行交互的。Applet 是能够在 Web 网页中运行的应用程序,它具有安全、功能强和跨平台等特性。

Applet 主要完成视频播放的功能。Applet 的程序代码是保存在 Web 服务器上的,当 Web 用户通过浏览器请求视频监控要求时,该程序下载到 Web 用户浏览器上运行。用户通过该程序进行现场动态监控。Applet 程序是由 Java 语言实现的。Applet 运行时首先调

用 init() 函数进行一些初始化工作,然后自动调用 start() 与视频服务器建立连接,将视频图像显示在 Web 用户的浏览器上。

3.3 Web 网页设计

Web 网页设计是用 HTML 语言设计的。HTML 语言是一种超文本标记语言,它包含了文档数据和显示格式两个部分,其中文档数据是显示在浏览器中数据内容,显示格式则规定了这些内容在浏览器中以格式呈现给用户的。HTML 能够用来表示超文本消息、邮件、文档、超媒体和数据库查询等,以及有内嵌图形的简单结构文档,还有已存在的信息的超文本视图。当用户请求 Web 网页时,浏览器就把 HTML 的数据显示给用户。

本系统网页的设计主要是通过 < applet > 标记的方式将 Applet 引入到 HTML 文档中。在 < applet > 和 < /applet > 之间标记一些参数属性。这个标记之间的程序代码设计如下。

```

< applet codebase = "." archive = "JWebcamPlayer.jar" code = "JWebcamPlayer.class"
name = "JWebcamPlayer" align = "center" width = "320" height = "240" MAYSRIPT >
< param name = "Color" value = "#ffffff" >
< param name = "Server" value = "127.0.0.1" >
< param name = "Port" value = "7070" >
< PARAM NAME = "scriptable" value = "true" >
< PARAM NAME = "mayscript" value = "true" >
< /applet >
  
```

其中 codebase 和 code 属性配合给出 Applet 类的完整路径,code 给出文件的名称,codebase 指出包含文件的路径。width 和 height 是 Applet 窗口的大小。align 是 Applet 窗口显示的位置。上述的这些代码完成了视频显示的主体工作。

4 实验测试

系统测试是在实验室内局域网进行的。客户端使用 RedHat Enterprise Linux AS4 操作系统,浏览器使用的是 Firefox 浏览器,由于服务器端的 Web 网页包含 Java 控件,默认安装的 Firefox 浏览器不支持 Java 插件,需要安装 Java 环境,让浏览器支持 Java 插件。硬件平台使用的是友善之臂 S3C2410 板。把本系统制作

出来的内核映像和文件系统映像下载到开发板。把摄像头插入到 USB 接口以及通过网线把开发板接入到本地局域网。重启开发板,通过终端把开发板 IP 设置为本地网址 202.193.74.220。运行 Web 服务器和视频服务器。客户端通过安装 JAVA 环境的 Firefox 浏览器输入 <http://202.193.74.220/webcam/webcam.html> 即可实现现场动态的监控。如图 5 所示实现监控本地实验室的情况。



图 5 网络视频监控实现图

5 结论

本系统的设计是在以数字信息技术和网络技术发展的前提下,运用当前的嵌入式技术和 ARM 技术设计网络视频监控系统,具有一定的创新性。该系统由于采用了高性能的 ARM9 处理器和嵌入式 Linux 操作系统组合,具有一定的可扩展性,可以通过适当的功能扩展来满足用户更高的要求。此外,可以针对不同的实际应用和硬件平台对 Linux 内核进行定制,以满足不同嵌入式平台用户的需要。本系统具有功能强,可靠性高,稳定性好和具有一定的扩展性特点,对嵌入式开发有一定的借鉴意义。

参考文献

- 1 郑灵翔. 嵌入式系统设计与应用开发. 北京:北京航空航天大学出版社, 2006.
- 2 毛德操,胡希明. LINUX 内核代码情景分析. 浙江:浙江大学出版社, 2001.
- 3 Jeremy Bentham. TCP/IP Lean Web Servers for Em-

bedded Systems 2E. 北京:机械工业出版社, 2003.

- 4 Adolfo Rodriguez, John Gatrell. TCP/IP Tutorial and Technical Overview 7th Edition. 北京:清华大学出版社, 2002.
- 5 赵方鹏,杨建华,等. 基于嵌入式 Linux 的网络视频监控. 测控技术, 2007, 5(26): 55-57.

(上接第 17 页)

```
CryptCreateHash( hProv, CALG_SHA1, 0, 0, &hHash );
//根据文件内容计算散列值,sourceFile 是一个文件流,pBuffer 是存放读文件内容的缓冲区,BUFFER 是缓冲区的大小
Do {
    dReadLen = sourceFile->Read( pBuffer, BUFFER );
}while( !dReadLen < BUFFER );
CryptHashData( hHash, pBuffer, dReadLen, 0 );
if( !CryptVerifySignature( hHash, signatureBytes, signatureBytesLen, hPublicKey, NULL, 0 ) )
{if( GetLastError( ) == NTE_BAD_SIGNATURE )
{ShowMessage( "文件已被修改" );拦截数据包;}
else
{ShowMessage( "文件没有被修改" );
允许数据包通过防火墙;}}
```

参考文献

- 1 Daniel Wan. Distributed Firewall. 2001.
- 2 朱雁辉. Windows 防火墙与网络封包截获技术. 北京:电子工业出版社, 2002:101-221.
- 3 北京启明星辰信息技术有限公司. 防火墙原理与实用技术. 北京:电子工业出版社, 2002:151-245.
- 4 张云勇,刘锦德. 移动 Agent 技术. 北京:清华大学出版社, 2003:51-201.
- 5 王红,曾广周,林守勋. 一种高效的移动 Agent 控制机制. 计算机工程和应用, 2002(2): 250-252.
- 6 周健. 基于移动代理的网络管理框架研究. 计算机应用, 2002(5): 48-50.
- 7 FARINACCI D, LI T, HANKS S. Generic Routing Encapsulation (GRE). RFC 2784, 2000.