

# 基于 ARM 的家居远程视频监控平台<sup>①</sup>

何少佳, 史剑清

(桂林电子科技大学 广西制造系统与先进制造技术重点实验室, 桂林 541004)

(桂林电子科技大学 机电工程学院, 桂林 541004)

**摘要:** 设计了一种基于嵌入式和 WIFI 无线网络的家用远程视频监控平台. 本系统采用基于 ARM11 内核的 S3C6410 微处理器作为控制器核心, USB 摄像头作为采集端, 利用 Video4Linux2(V4L2)提供的应用函数等完成视频采集, 使用 T264 软件进行编码解码, 采用 WIFI 无线网络, 依靠 RTP/ RTCP 协议和 TCP 协议实现视频数据和控制指令的远程传输. 实验结果表明, 本平台监控视频画面流畅, 能很好的满足家居无线视频监控的应用需求. 同时本平台结构简单、造价低廉、易于推广.

**关键词:** 家用视频监控; 嵌入式; V4L2; WIFI 无线传输; RTP/RTCP

## Household Remote Video Monitoring Platform Based on ARM

HE Shao-Jia, SHI Jian-Qing

(Guangxi Key Laboratory of Manufacturing System & Advanced Manufacturing Technology, Guilin University of Electronic Technology, Guilin 541004, China)

(College of Electromechanical Engineering, Guilin University of Electronic Technology, Guilin 541004, China)

**Abstract:** A household remote video monitoring platform is designed and basis of embedded and WIFI wireless network. This system chooses S3C6410 microprocessor which based on ARM11 cores as the kernel controler regards USB camera as capture terminal, captures video with application function which provided by Video4Linux2, encodes and decodes the video by T264 software, and transfers the video data and control instruction rely on RTP/RTCP protocol and TCP protocol through WIFI wireless network. The experimental results show that this platform makes video image smooth, and it can well meet the demand of household wireless video monitoring application. At the same time, this platform has simple structure, low cost, and easy to promote.

**Key words:** household remote video monitoring; embedded; V4L2; WIFI wireless transmission; RTP/RTCP

随着计算机技术、网络技术的快速发展, 智能家居很大程度上优化了人们的生活方式, 能有效帮助人们安排时间、节约能源, 而视频监控系统是智能家居的重要组成部分之一<sup>[1]</sup>. 传统的家居视频监控系统大多数是基于有线环境, 它们需要固定在某一固定位置, 即便是有云台控制技术, 也不能监控到每个房间每个角落. 为了解决这一问题, 基于无线网络的嵌入式视频监控系统成为家居监控领域研究的重点.

在过去, 家庭网络一般采用有线网络, 相对来说比较繁琐, 现在无线网络成为家庭网络的首选. 在应用

较多的无线网络中, 常用的网络标准主要有 WIFI、蓝牙、Zigbee、3G 等. 但是蓝牙传输距离较短, Zigbee 传输速率只有 250kb/s, 3G 网络成本的较高, 都不适用于家居视频监控.

根据家庭内部网络的特点, 本文设计一种基于 WIFI 网络技术和嵌入式的可移动家居视频监控平台. 采用 USB 摄像头作为采集端, 利用 V4L2 提供的应用函数等完成视频采集, 采用具有高压缩率的 H.264 算法进行视频压缩, 利用 WIFI 无线网络, 依靠 RTP/RTCP 协议和 TCP 协议实现视频数据和控制指令

① 基金项目:广西制造系统与先进制造技术重点实验室课题(0842006\_020\_Z)

收稿时间:2013-12-09;收到修改稿时间:2014-01-20

的远程传输,把云台技术与智能小车结合起来,方便查看各个房间信息.

### 1 系统整体设计

本系统以移动小车为载体,分上下两层.上层放置主控制核心、USB 摄像头及其他外围模块,其中 USB 摄像头与一个舵机相连,可控制舵机带动摄像头上下运动,置于机车顶端,用于摄取图片、视频;下层放置单片机及电机驱动模块.

软件部分主要是针对 Linux 系统的配置、整合.为了实现监控功能,还需要开发监控终端以获取所需数据及实现监控.具体结构框架如图 1 所示.

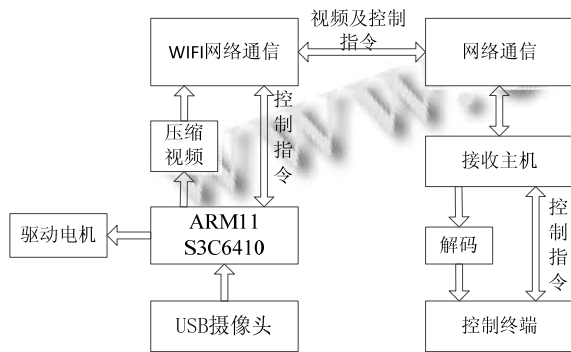


图 1 系统总体框图

### 2 系统硬件设计方案

本系统的硬件主要由控制核心、视频采集模块、数据传输模块、单片机控制模块、电机驱动模块、SDRAM、SD 卡等外围电路模块组成,系统硬件结构图如图 2 所示.

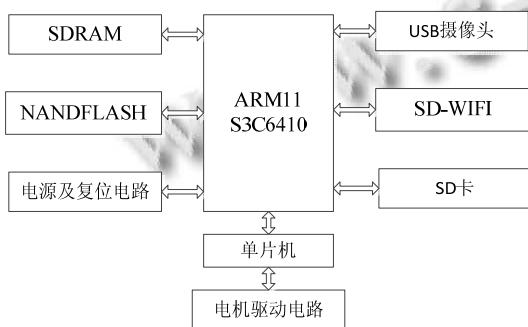


图 2 硬件结构图

本系统设计采用两级处理方案:一级为 ARM11 微处理器,主要负责视频采集、压缩和传输等相关任务;二级处理器为高速单片机,与主控制核心相连,

接收 ARM 解析的控制数据,负责控制电机驱动电路.

其中,ARM11 微处理器是采用三星公司的 S3C6410 处理器,该处理器集成多格式编解码器可以支持多种视频格式的编解码,并且编码速度快,能够满足实时视频监控的要求;采用体积小、成本低的 USB 摄像头采集视频信号,调用集成在处理器内部的 V4L2 提供的应用函数完成摄像头数据的采集;通过 SD-WIFI 模块建立网络传输.在此基础上,控制核心板配置了 1GB 的 NANDFLASH 和 128MB 的 SDRAM 内存, NANDFLASH 用于存储 Linux 操作系统的内部的数据和应用程序, SDRAM 用于存放运行程序及摄像头所采集的数据.

### 3 系统软件设计方案

#### 3.1 搭建操作系统平台

嵌入式 Linux 开发环境一般是由 Bootloader 引导、Linux 操作系统、用户应用程序及相关硬件组成. Bootloader 相当于 Windows 操作系统里的 Bios,可以初始化硬件设备、分配内存空间,从而使系统的软硬件环境达到合适状态,为调用操作系统内核做好准备. Linux 操作系统是一种开源系统,可移植性强、定制裁剪方便,主要由内核与根文件组成,本文选取 Linux2.6.36 作为内核, cramfs 作为根文件,方便对代码进行配置裁剪并能有效节省内存空间.本系统的软件构架主要有视频采集压缩程序、数据传输程序、控制程序,图 3 是软件系统结构图.

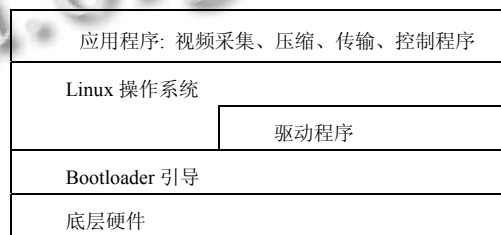


图 3 软件系统结构图

#### 3.2 视频数据采集

本系统选用 Video4Linux2(V4L2)来实现对视频设备的各种控制. V4L2 是嵌入在 Linux 内核里的支持视频设备的应用程序接口,在 Linux 操作系统下,配合视频设备及其驱动程序,可以方便的设置摄像头的频率、帧频、视频压缩格式和图像参数等<sup>[2]</sup>. V4L2 在 Linux 操作系统下进行视频采集的函数主要有 open、

close、read、write、ioctl、select、mmap 等,这与普通字符设备的驱动接口在形式上相一致的,但是 V4L2 可以利用 ioctl 函数提供的丰富的控制字来实现对设备的管理,包括对设备信息的查询、缓冲的设置、图像属性的设置、视频的捕捉等。

本系统视频采集程序流程如下:

- ① 使用 open 函数打开设备文件;
- ② 调用 vidioc\_querycap 来获取设备的性能参数;
- ③ 根据 ioctl 函数返回图像的属性,获得图像参数,将这些参数存放到 video\_picture 中;
- ④ 采用 mmap 映射方式读取图像数据.首先在内核空间申请多个缓冲区,然后将每个缓冲区通过 mmap 映射到用户空间,使驱动和应用程序共享这些缓冲区,这样在进行数据处理时就不用进行拷贝,大大提高效率.
- ⑤ 对采集的视频图像进行压缩或者变换处理,本系统采用的是 H.264 视频压缩标准;由于选用的摄像头输出的格式是 RGB24,为了节省存储空间,需将其转换为 YUV420 格式之后在进行压缩编码;
- ⑥ 使用 close 函数关闭视频设备,解除对摄像头的控制.

视频数据采集程序流程图如图 4.

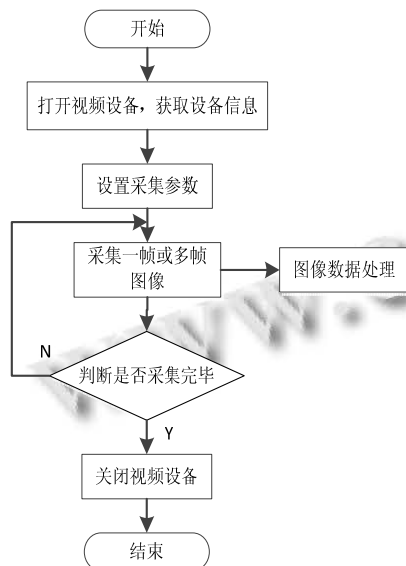


图 4 视频数据采集程序流程图

### 3.3 基于 H.264 视频数据压缩

H.264 是一种高性能的视频编解码技术,是由国际电联(ITU-T)和国际标准化组织(ISO)的运动图像

专家组联合定制的视频编码标准<sup>[3]</sup>.与其他标准相比,H.264 具有更高的图像压缩率和图像质量,已经成为现在主流的视频处理协议,广泛应用在实时视频监控、互联网视频传输、视频数据存储等方面。

目前,针对 H.264 的编解码器软件有很多,主流的开源编解码器有:JM、X264、T264 等.本文使用的是 T264 方案.

T264 是由中国视频编码自由组织联合开发的 H.264 编解码器,它遵循 H.264 编码准则,并吸收了 JM、X264、Xvid 三个源码的优点<sup>[4]</sup>.本文采用的 T264 编码库版本为 avc-src-0.14,里面的源代码主要集中在 common、decoder、encoder 和 dshow 目录下,而 build 目录下存放的是 Linux 系统平台下对整个视频编码函数的构建文件,并且 Linux 目录下 T264 的视频压缩编码主线函数均在 makefile 内构建好了.进入 T264 编码库的“avc-src-0.14/avc/built/linux”目录下,修改 makefile 文件,使用 vi 修改其中 CC 与 CPP 等,通过 make 命令编译即可,编译后的可执行文件 t264 位于 T264 编码库的“avc-src-0.14/avc/build/obj”目录下.

先对 T264 编码环境初始化设置,然后就可以对转换后的 YUV420 图像数据进行压缩编码.

### 3.4 视频数据传输程序

压缩后的视频数据通过家庭建立好的 WIFI 网络与 Internet 进行连接.本系统中视频数据的传输是基于实时传输协议 RTP/RTCP. RTP 是一种实时传输协议,能为多目标广播网络或单目标广播网络提供一对一的实时传输服务,位于 TCP 和 UDP 协议层之上<sup>[5]</sup>. RTCP 是一种实时传输控制协议,其主要功能是监控、反馈服务质量,实现视频音频的同步以及多组成员的识别.

视频数据传输程序主要负责将压缩后的数据信息封装成 RTP 包,并不断发射、接收 RTCP 包进行反馈控制.本系统中,采用的是对独立 NAL 单元进行封装的方案,首先将压缩数据封装成 RTP 包,然后将其封装成 UDP 包,最后封装成 IP 数据包在网络中传输. IP 数据包通过网络传输到接收端,接收端将收到的 IP 数据包按照相反的顺序把 RTP 报头和压缩数据提取出来,然后根据 RTP 报头中的编号把压缩数据输入到接收端的解码器进行解码输出.图 5 为视频传输具体过程.

本系统中视频数据传输是基于 JRTPLIB 库来实现 RTP/RTCP 协议的,JRTPLIB 库是基于 C++ 的开源 RTP 库,封装性好,操作简单,支持 Windows、Linux 等多种平台<sup>[6]</sup>.下载 JRTPLIB-3.7.1 和 JTHRED-1.2.1 到

Linux 操作系统上, 先对 JTHRED 进行编译安装, 再安装 JRTPLIB. 由于开发应用程序时用到的是动态链接方式编译, 还需要将生成的 JRTPLIB 库拷贝到/usr/lib 目录下.

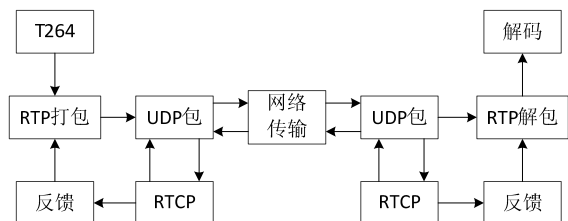


图 5 视频传输过程

利用 JRTPLIB 库提供的接口实现视频传输的流程如图 6.

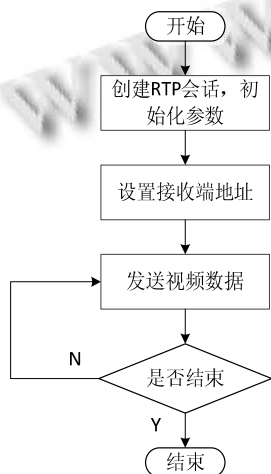


图 6 JRTPLIB 实现视频传输流程

主要实现步骤如下:

① 创建 RTP 会话

```

RTPSession sesses;//定义会话
RTPUDPv4TransmissionParams transparams;
RTPSessionParams sessparams;
transparams.SetPortbase(portbase); //设置会话端口号
    
```

号

```

sessparams.SetOwnTimestampUnit(1.0/50.0);//设置发送时间间隔
    
```

```

status=sesses.Create(sessparams, &transparams);//
    
```

初始化会话

```

checkerror(status);//判断初始化是否成功
    
```

② 添加接收端地址

```

Unsigned long destip=inet_addr(192.168.23.10);
    
```

```

Unsigned short destport=8000;
    
```

RTPIPv4Address addr(destip, destport);//指定目的端地址和端口号

```

status=sesses.AddDestination(addr);//添加接收数据地址
    
```

```

checkerror(status);//判断是否成功
    
```

③ 数据发送

```

status=sesses.SendPacket((void*)buf,1000);
    
```

④ 数据接收

在 RTP 协议中允许一个会话接收多个数据源的数据, 可以使用 GotoFirstSourceWithData() 和 GotoNextSourceWithData() 来遍历所有数据源的数据.

3.5 控制系统实现

本系统的控制部分主要体现为控制终端对移动小车和摄像头的控制. 通过点击控制终端界面的控制球, 可实现控制小车前后左右移动和摄像头的上下移动.

为确保控制指令的正确、可靠传输, 因此对舵机的控制采用的是 TCP/IP 协议. TCP 协议需要客户端和服务端进行三次握手, 以保证数据传输的可靠性: 在数据传输前需要先建立逻辑连接, 然后再进行数据的传输, 最后释放连接. 控制指令由控制终端的 OutputStream 输出, 由本地系统的 InputStream 接收, 控制数据通过 Socket 通信的实现方式如下:

控制终端:

```

Socket socket = new Socket("192.168.23.12",4567);
    
```

//创建一个 Socket 对象, 指定 IP 地址和端口号

```

InputStream inputStream = new
    
```

```

FileInputStream(File file); //使用 InputStream 读取数据
OutputStream
    
```

```

outputStream=socket.getOutputStream();
    
```

```

byte buf [] = new byte[];
    
```

//从 Socket 中得到 OutputStream

```

int temp = 0;
    
```

```

inputStream.read(buf); //将 InputStream 当中的的数据读出
    
```

```

outputStream.write(buf,0,temp);//写入
    
```

OutputStream

本地系统:

```

ServerSocket serverSocket = new
    
```

ServerSocket(4567); //创建 ServerSocket 对象, 用以监

听 4567 端口

```
Socket socket = serverSocket.accept();//接收控制终端发送的请求
```

```
InputStream inputStream = socket.getInputStream();//从 Socket 中得到 InputStream 对象
```

```
Byte buf [] = new byte[];
```

```
inputStream.read(buf); //将指令存入 buf
```

通过以上主要程序就可以实现控制指令的网络传输功能, 最终实现控制终端对移动平台的控制。

#### 4 系统测试与结论

本文设计了一种基于嵌入式和 WIFI 无线网络的家用远程视频监控平台。该系统采用以 ARM11 为内核的 S3C6410 处理器作为核心, 实现利用 V4L2 技术对 USB 摄像头进行数据采集, 并进行 H.264 编码, 压缩后的数据采用 RTP 封装打包, 经 WIFI 无线网络与远程监控客户端进行交互。

本系统在实际网络中进行了测试, 采用三星公司的 S3C6410 开发板进行实验, 在远程 PC 端用 VC 完成控制终端测试程序, 网络环境为 WIFI 无线局域网, 运行效果如图 7 所示。在测试中, 设置图片大小为 352\*288, 每秒钟的图片帧数为 15 帧, 视频图像清晰、流畅, 能满足家庭视频监控的实际需求。

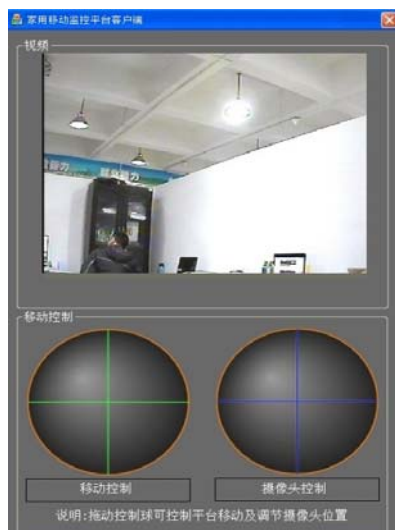


图 7 PC 控制终端

#### 参考文献

- 1 王家良,王朝晖,焦斌亮.视频监控系统在家居方面的应用.电视技术,2008,32(5):94-96.
- 2 冯国进.Linux 驱动程序开发实例.北京:机械工业出版社,2011.
- 3 张多英,申晨,刘伟平,黄红斌.嵌入式视频监控传输系统的设计与实现.计算机工程与设计,2010,31(4):724-728.
- 4 张前进.基于 T264 的 H.264 视频采集与处理的设计与实现.中国西部科技,2011,10(27):28-29.
- 5 李燕灵,马瑞芳,左力.基于 RTP/RTCP 的实时视频数据传输模型及实现.微电子学与计算机,2005,22(8):138-140.
- 6 孙克辉,尧平,洪娟娟,喻炜.基于 JRTPLIB 库的 H.264 视频传输系统.计算机系统应用,2011,20(12):21-24.