

# 基于 OMAP 远程视频监控系统的研究与实现

陈晓霜, 吴 蒙

(南京邮电大学, 江苏 南京 210003)

**摘 要:**文中提出了一种基于 TI 的 OMAP4460 多媒体平台,利用 V4L2 视频采集框架,结合 H. 264 压缩编码技术,采用 RTSP 流媒体技术实现远程视频监控系统的解决方案。通过软硬件平台的搭建,对视频原始数据采集过程的分析和实时视频网络传输的实现,文中较全面地展示了远程视频监控系统的实现过程。通过实际运行验证了该嵌入式远程监控系统的可行性。系统测试结果表明,该系统能够及时准确地采集现场视频图像信息,并通过以太网将采集到的数据远程实时传输,在客户端实时显示,具有实时、稳定的特点,满足了实时数据传输和远程监测的需求。

**关键词:**视频监控;流媒体;嵌入式 Linux;压缩编码

中图分类号:TP302.1

文献标识码:A

文章编号:1673-629X(2015)04-0177-04

doi:10.3969/j.issn.1673-629X.2015.04.040

## Research and Implementation of Remote Video Monitoring System Based on OMAP

CHEN Xiao-shuang, WU Meng

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

**Abstract:** A kind of solution based on TI OMAP4460 multimedia platform is put forward, making full use of the V4L2 video collection framework, combined with H. 264 compression coding technology and adopted RTSP streaming technology to realize the remote video monitoring system. Through the construction of the software platform and hardware platform, the analysis of the original video data acquisition process and the realization of real-time video remote transmission, comprehensively show the implementation of the network video monitoring system. The experiment has verified the feasibility of the embedded remote monitoring and control system. Results show that the system can collect the live video information timely and accurately, besides, the collected data can be transmitted through the Ethernet to client successfully. The real-time and stability of the system meets the needs of real-time data transmission and remote monitoring.

**Key words:** video monitoring; streaming media; embedded Linux; compression coding

## 0 引言

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<sup>[1]</sup>。随着计算机、网络、通信技术的日趋成熟,各种实用型视频技术的不断完善,以嵌入式技术为依托,以网络、通信技术为平台,以智能图像分析为特色的网络视频监控系统使视频监控从幕后走向了前台。

H. 264 是一种高性能的视频编解码技术<sup>[2]</sup>,它最大的优势是具有很高的数据压缩比率,H. 264 不仅具有高压比,还拥有高质量流畅的图像。此外,H. 264

还支持低延时工作,以便支持实时通信应用,同时提供检错纠错功能,对丢失包和误码误比特进行处理,自动检错重发<sup>[3]</sup>。

V4L2 (Video For Linux Two) 是内核提供给应用程序访问音、视频驱动的统一接口<sup>[4]</sup>。V4L2 主要功能是实现设备发现和操作设备的能力。mmap() 和 ioctl() 是 V4L2 在视频采集中的两个重要系统调用,mmap() 系统调用使得进程之间通过映射同一个普通文件来实现共享内存;ioctl() 系统调用对设备 I/O 通道进行控制。共享内存方式使系统的效率得到大大的提高<sup>[5]</sup>。

收稿日期:2014-06-07

修回日期:2014-09-10

网络出版时间:2015-02-23

基金项目:国家“973”重点基础研究发展计划项目(2011CB302900);江苏省高校自然科学基金研究重点项目(10KJA510035);南京市科技发展计划重大项目(201103003)

作者简介:陈晓霜(1989-),女,硕士研究生,研究方向为无线通信与信号处理;吴 蒙,教授,研究方向为无线通信与信号处理技术、无线网络安全与通信系统的信息安全。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20150223.1239.030.html>

实时流传输协议 (Real Time Streaming Protocol, RTSP), 是 TCP/IP 协议体系中的一个应用层协议, 控制实时数据的传送, 获得了广泛支持。该协议定义了一对多应用程序有效地通过 IP 网络传送多媒体数据。RTSP 是双向的, 即使用 RTSP 时, 客户机和服务器都可以发出请求。RTSP 可以充当多媒体服务器的网络远程控制<sup>[6]</sup>。

基于 OMAP4460<sup>[7]</sup> 处理器, 通过使用 RTSP 协议, 结合 H. 264 多媒体压缩标准的监控系统, 充分解决了通过网络进行实时画面监控的问题。文中主要针对的就是基于 OMAP4460 的远程视频监控系统的实现。

## 1 网络监控系统总体概述

该系统由视频采集模块、数据处理模块、视频显示模块三大部分组成。视频采集模块由 e-CAM51\_44x 摄像头子模块、摄像头驱动程序 V4L2 两部分构成。数据处理模块由 H. 264 编码和网络传输应用程序组成。视频显示模块由运行于 gstreamer 环境之上的客户端程序组成, 系统结构如图 1 所示。

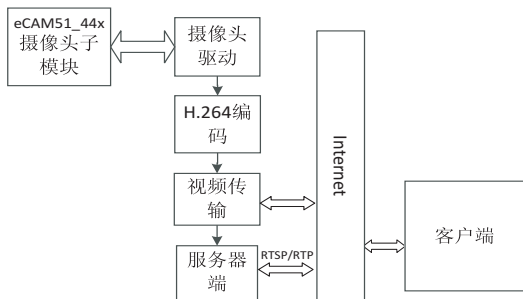


图 1 系统总体结构图

## 2 系统硬件设计

该系统搭建在 PandaBoard 开发板上。PandaBoard 主处理器是 TI 的 OMAP4460, OMAP (Open Multimedia Application Platform) 是美国德州仪器公司的开放式多媒体应用平台, 它是 TI 公司推出的第三代移动通信的开发平台<sup>[8]</sup>。

OMAP4460 双核处理器主频达到 1.2 GHz, OMAP4460 搭载了 CortexA9MP 架构的双核心, 1 MB 二级缓存, 采用 45 nm 工艺。内存方面支持双通道 LPDDR2 1066 的内存, 而 GPU 采用 PowerVR SGX540, 频率高达 384 MHz。OMAP4460 处理器是一个功能强大的系统芯片, 电源效率和高性能完美平衡。它支持 HDMI1.4 接口 3D 高清视频输出, 支持 1080p 60FPS 高清解码, 支持最高 1 200 万像素双摄像头, 集成 IVA3 硬件加速视频解码器。文中的监控系统便是搭载 OMAP4460 处理器上的。图 2 是文中监控系统的总体硬件图。

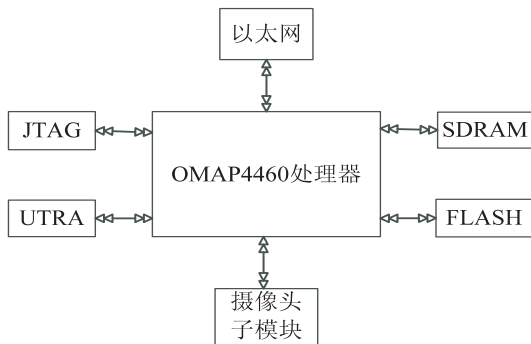


图 2 监控系统硬件图

## 3 系统软件设计

### 3.1 嵌入式平台的搭建

#### 3.1.1 uboot 移植

uboot 可支持多平台多操作系统, 其功能是在操作系统内核启动前运行一段小程序。通过这段程序, 可以初始化硬件设备, 为最终调用操作系统内核做好准备, 并将内核映像拷贝到内存当中, 跳到入口地址启动内核<sup>[9-10]</sup>。在对 uboot 移植之前, 应先安装交叉编译工具链, 文中选择 arm-linux-gcc-4.4.3 版本的交叉工具链, uboot 的版本为 2008.10, 在对 uboot 的移植过程中应保留其对 NOR flash、串口、USB 和网卡的支持。按要求修改好各配置文件后, 执行下面命令

```
make OMAP4460panda_config
make CROSS_COMPILE=arm-linux-
```

在顶层目录下生成 u-boot. bin 文件, 将该文件烧写到开发板上能够正常运行。

#### 3.1.2 Linux 内核的移植

Linux 内核和根文件系统是组成嵌入式 Linux 系统的两部分。因此相应的 Linux 内核与根文件系统的成功移植才能实现 Linux 内核的完整移植<sup>[11]</sup>。内核制作主要是:

(1) 用指令 make distclean 来清除之前的配置和中间文件;

(2) 用命令 make menuconfig ARCH = arm 来配置内核;

(3) 用指令 make uimage ARCH = arm 实现内核编译。

分 7 个步骤来制作根文件系统: 根文件系统目录的创建, 设备文件的创建, 内核模块的安装和编译, busybox 的配置、编译和安装。

### 3.2 V4L2 实现视频的采集

(1) 打开视频设备。

```
int fd = open (/dev/video0, O_RDWR | O_NONBLOCK);
```

使用非阻塞模式调用视频设备, 即使尚未捕获到

信息,驱动依旧会把缓存(DQBUF)里的东西返回给应用程序。文中采用非阻塞模式。

### (2)能力查询。

使用 ioctl VIDIOC\_QUERYCAP 来查询当前 driver 是否合乎规范。因为 V4L2 要求所有 driver 和 Device 都支持这个 ioctl。所以,可以通过这个 ioctl 是否成功来判断当前设备和 driver 是否支持 V4L2 规范。当然,这样同时还能够得到设备足够的能力信息。

```
struct v4l2_capability {
    __u8 driver[16]; //驱动名
    __u8 card[32]; //Device 名
    __u8 bus_info[32]; //Bus 系统中存放位置
    __u32 version; //driver 版本
    __u32 capabilities; //能力集
    __u32 reserved[4];
};

int ioctl(int fd, int request, struct v4l2_capability *
argp);
```

### (3)设置视频捕获格式。

设置图片的格式为 YUV,单帧图片的大小为1 080 \* 720,主要代码如下:

```
fmt.fmt.pix.height = 720;
fmt.fmt.pix.width = 1 080;
fmt.fmt.pix.field = V4L2_FIELD_INTERLACED;
fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_YUYV;
if(ioctl(fd, VIDEO_S_FMT) == -1) { return -1; }
```

### (4)Streaming 信息设置。

前面 Format 是单帧数据的设置,现在设置和流有关的信息,如帧数。这里帧数设置为 30 fs。

```
Struct v4l2_streamparm stream_parm;
memset(&stream_parm, 0, sizeof(stream_parm));
Stream_Parm.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
```

```
Stream_Parm.parm.capture.timeperframe.denominator = 30;
```

```
Stream_Parm.parm.capture.timeperframe.numerator = 1;
```

```
io_rel = ioctl(Handle, VIDIOC_S_PARM, &Stream_Parm);
```

### (5)分配内存。

结构体 V4l2\_requestbuffer 为驱动进行视频缓存数量的申请提供依据,一般向驱动申请的帧缓冲不超过 5 个。

```
Struct v4l2_requestbuffers req;
ret = ioctl(fd, VIDIOC_REQBUFS, &req);
//获取了 req.count 个缓存
```

```
ret = ioctl(fd, VIDIOC_QUERYBUF, &buf);
//获取这些缓存地址
mmap(NULL, buf.length, PROT_READ |
PROT_WRITE, MAP_SHARED, fd);
//mmap 函数转换成应用程序中的绝对地址
ret = ioctl(fd, VIDIOC_QBUF, &buf);
//申请的缓冲进入队列
```

### (6)捕捉视频<sup>[12]</sup>。

将已分配的 buffer 调用 VIDIOC\_QBUF 将 buffer 全部加入输入队列中,并调用 VIDIOC\_STREAMON 开始捕获数据:

```
ret = ioctl(fd, VIDIOC_STREAMON, &type);
```

其中:

```
type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
```

### (7)取出缓存帧数据。

利用 buf.index 的返回值查找相应的 mmap 映射缓存后将视频数据取出。

```
if(ioctl(fd, VIDIOC_DQBUF, &buf) == -1)
{ return -1; }
```

### (8)循环采集。

开始捕获数据以后就会进入一个主循环,可以使用 select 或者 poll 来监听文件描述符的状态,一旦有数据可读,就调用函数来读取数据。文中采用 Memory mapped 方式。Memory mapped 方式下,先从输出队列中 dequeued 一个 buffer,然后将处理完成的帧数据放入输入队列,由此视频数据的采集便实现了。

```
if(ioctl(fd, VIDIOC_QBUF, &buf) == -1) { return -1; }
```

## 3.3 视频网络传输模块设计

OPTIONS、DESCRIBE、SETUP、PLAY 以及 TEARDOWN<sup>[13]</sup>是视频网络传输整个过程中使用的 RTSP 协议的主要方法。OPTIONS 用来获取服务器/客户端支持的能力集;DESCRIBE 用来从服务器获取流媒体文件格式信息和传输信息;SETUP 用来与服务器协商流媒体传输方式,此过程中,建立 RTP 通道;PLAY 用来与服务器协商流媒体播放;TEARDOWN 用来拆除连接。

具体实现流程如图 3 所示。

具体过程如下:

(1)客户端发出 RTSP 请求报文 OPTION,RTSP 服务器端收到此报文后,返回 REPLY 报文作为回应;

(2)客户端向服务器端发送 DESCRIBE 请求报文用于得到 URI 所指定的媒体描述信息,服务器端收到 DESCRIBE 请求报文后,给出回应报文,在回应报文中,给出了 SDP 信息;

(3)客户端发出 SETUP 报文请求用于确定传输机

制,建立 RTSP 会话。这样,RTSP 客户端和服务端就建立起 RTSP 会话;

(4) RTSP 客户端发出 PLAY 报文请求,PLAY 方法通知服务器利用 SETUP 中指定机制开始发送数据。如果还未收到 SETUP 请求的成功应答,客户端不允许发出 PLAY 请求。

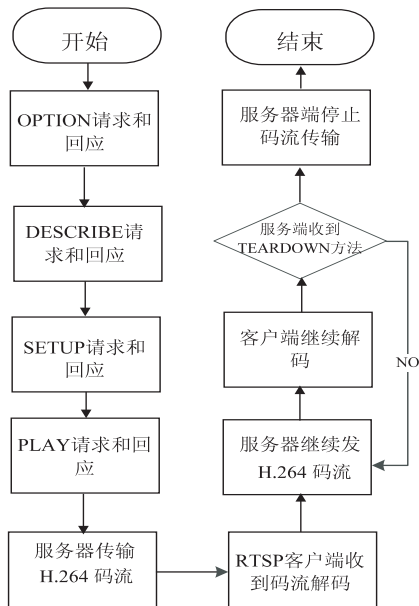


图 3 RTSP 实现 H. 264 码流传输流程图

在完成以上 4 个报文请求和回应后,RTSP 服务器就开始传输 H. 264 视频流,客户端根据收到的码流开始 H. 264 解码,这样整个 H. 264 视频直播系统就构建完成了。客户端给服务器端发出 TEARDOWN 报文后,服务器停止传输 H. 264 码流。

### 4 系统测试与分析

文中的视频监控系统分为客户端(Client)和服务端(Server),Client 程序的任务是将用户的要求提交给 Server 程序,再将 Server 程序返回的结果以特定的形式显示给用户;Server 程序的任务是接收客户程序提出的服务请求,进行相应的处理,再将结果返回给客户程序<sup>[14]</sup>。文中 Client 程序是笔者编写的 send\_c,用于请求服务端传输实时视频信息,Server 程序是 receive\_c,用于响应客户端请求,并实现实时视频的网络传输。测试时,由于实验所在网段是 10. 10. 210, PandaBoard 开发板与显示器连接成功后,通过键盘进入系统,在终端输入 sudo. /ne\_send\_c 10. 10. 201. 255,打开服务端。接着在客户端输入 ./receive,以接收从服务端返回的实时视频数据。图 4 展示了服务端和客户端的交互情况。

具体测试的过程如下:

(1) 将 PandaBoard 开发平台与鼠标、键盘、显示屏连接,并用网线将开发板与有线网络连接,通过显示屏

查看开发平台的 IP 地址信息,显示的 IP 地址为 10. 10. 201. 42。

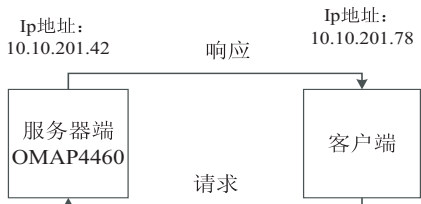


图 4 服务端和客户端交互图

(2) 在终端输入 sudo. /ne\_send\_c 10. 10. 201. 255, 打开服务端。

(3) 在客户端输入 ./receive,以接收从服务端返回的实时视频数据。

客户端接收视频结果如图 5 所示。



图 5 客户端接收视频结果

### 5 结束语

文中提出了一种基于 OMAP4460 处理器,利用 V4L2 视频采集框架,结合 H. 264 压缩编码技术,采用 RTSP 流媒体技术实现远程视频监控系统的解决方案,实现了视频的网络实时传输并解决了在低带宽限制情况下传输高质量图像信息的难题,具有很好的发展前景。

#### 参考文献:

[1] 付 蓉,严建亮. 智能家居远程视频监控系统的设计与实现[J]. 计算机技术与发展,2012,22(3):137-140.

[2] 毕厚杰. 新一代视频压缩编码标准—H. 264/AVC[M]. 北京:人民邮电出版社,2005.

[3] Vun N,Ansary M. Implementation of an embedded H. 264 live video streaming system[C]//Proc of IEEE 14th international symposium on consumer electronics. Braunschweig: IEEE, 2010:1-4.

[4] 冯国进. Linux 驱动程序开发实例[M]. 北京:机械工业出版社,2011:327-333.

[5] Liu Sheng,Zhao Jingjing,Fan Xiuli. The embedded video surveillance system based on V4L2[J]. Microcomputer Applica-



入和断开-接入等类型验证项,对各项功能进行充分验证。

FPGA 原型验证平台为某编码系统的验证平台,由 AD 芯片、处理器芯片、FPGA 芯片组成,如图 6 所示。

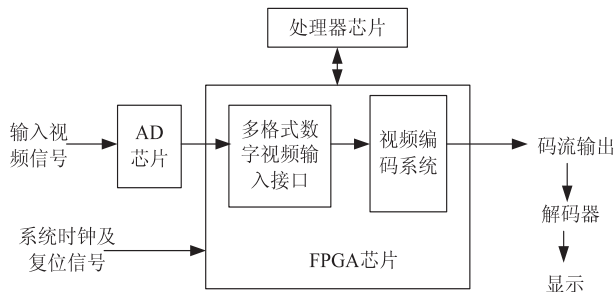


图 6 FPGA 原型验证平台

其中,FPGA 芯片中烧录多格式数字视频输入接口和视频编码系统的逻辑综合。来源于视频发生设备的输入视频信号通过 AD 芯片转换为数字信号输入给多格式数字视频输入接口,采集得到的视频数据输出给视频编码系统进行编码,输出码流经解码器解码后由显示设备显示<sup>[13-14]</sup>。通过观察显示效果分析多格式数字视频输入接口的正确性。

## 4 结束语

针对视频编码系统对前端视频输入接口多格式、鲁棒性和容错功能的需求,深入剖析各种标准视频格式的特征及相关性,采用滤波和同步采样、统一视频格式、数据采集状态机和行像素数容错等设计方法,设计出一种多格式数字视频输入接口,支持 TV 格式和 VESA 格式的多种分辨率视频,具有高鲁棒性和容错能力。经过虚拟平台验证和 FPGA 原型验证,证明该设计的功能和性能均达到设计要求,可应用于视频编码系统。

## 参考文献:

- [1] Jack K. Video demystified[M]. 5th ed. [s. l.]:Elsevier Inc, 2007:15-25.
- [2] H. 264 codec LSI MB86H51 product specification[S]. 2008.
- [3] VESA Video Interface Port (VIP)[S]. 1998.
- [4] VESA monitor timing standard, VESA and industry standards and guidelines for computer Display Monitor Timing (DMT)[S]. 2004.
- [5] Interfaces for digital component video signals in 525-line and 625-line television systems operating at the 4:4:4 level of recommendation ITU-R BT. 601 (Part A)[S]. ITU-R BT. 799-3, 1998.
- [6] Interfaces for digital component video signals in 525-line and 625-line television systems operating at the 4:2:2 level of recommendation ITU-R BT. 601[S]. ITU-R BT. 656-4, 1998.
- [7] Television-1920×1080 image sample structure, digital representation and digital timing reference sequences for multiple picture rates[S]. SMPTE 274M-2003, 2003.
- [8] Television-1280×720 oggressive image sample structure, analog and digital representation and analog interface[S]. SMPTE 296M-2001, 2001.
- [9] Benson K B. Television engineering handbook[M]. [s. l.]: McGraw-Hill, Inc, 1986.
- [10] EBU interfaces for 625-line digital video signals at the 4:2:2 level of CCIR[S]. EBU Tech 3267-E, 1992.
- [11] 夏宇闻. Verilog 数字系统设计教程[M]. 北京:北京航空航天大学出版社, 2003:304-305.
- [12] 田 靖, 田 泽. AFDX-ES SoC 虚拟仿真平台的构建与应用[J]. 计算机技术与发展, 2010, 20(8):192-194.
- [13] 毕厚杰. 新一代视频压缩编码标准—H. 264/AVC[M]. 北京:人民邮电出版社, 2004:21-25.
- [14] IEC 61883-1, 2003, consumer audio/video equipment—digital interface—part 1: general[S]. [s. l.]: European Broadcasting Union, 1991.

(上接第 180 页)

- tions, 2011, 32(1):37-42.
- [6] 朱伟雄. 新一代数据中心建设理论与实践[M]. 北京:人民邮电出版社, 2009:188-194.
- [7] OMAP4460 multimedia device silicon revision 1. x, revision AB[M]. [s. l.]:Instruments TI, 2012.
- [8] OMAP4430 processors[EB/OL]. [2013-02-18]. <http://www.ti.com/general/docs/wtbu/wtbuproducentcontent.tsp?contentId=53243&navigationId=12843&templateId=6123>.
- [9] Zhang Lefang, Wang Jianxin, Zhang Kai. Design of embedded video monitoring system based on S3C2440[C]//Proc of fourth international conference on digital manufacturing & automation. Qingdao:IEEE, 2013:461-465.
- [10] 申 爽. 基于 S3C2440 的 Uboot 分析与移植[J]. 计算机系

统应用, 2012, 21(5):222-225.

- [11] Hu Jie, Zhang Genbao. Research transplanting method of embedded Linux kernel based on ARM platform[C]//Proc of international conference of information science and management engineering. Xi'an:IEEE, 2010:35-38.
- [12] Kuang Shunming, He Xiaojian. Design and application of CMOS device driver based on S3C2440[C]//Proc of the tenth international conference on electronic measurement & instruments. Chengdu:IEEE, 2011:110-114.
- [13] 张功国, 陈莹星. 实时视频中 SIP 与 RTSP 的应用[J]. 电视技术, 2013, 37(9):116-118.
- [14] 刘 媛, 张 伟, 王知学. 基于 B/S 和 C/S 架构的嵌入式远程监控系统[J]. 仪表技术与传感器, 2008(10):39-41.

# 基于OMAP远程视频监控系统的研究与实现

作者：[陈晓霜](#)，[吴蒙](#)，[CHEN Xiao-shuang](#)，[WU Meng](#)

作者单位：[南京邮电大学](#), [江苏 南京](#), 210003

刊名：[计算机技术与发展](#)

英文刊名：[Computer Technology and Development](#)

年，卷(期)：2015(4)

引用本文格式：[陈晓霜](#), [吴蒙](#), [CHEN Xiao-shuang](#), [WU Meng](#) [基于OMAP远程视频监控系统的研究与实现](#)[期刊论文]-[计算机技术与发展](#) 2015(4)