

基于 H.264 的嵌入式视频编码器的设计与应用

范新南, 邢 超

(河海大学常州校区计算机及信息学院, 常州 213022)

摘 要: 设计了一种适用于工业现场的嵌入式视频编码器, 视频压缩按照 H.264 标准, 在 Intel 推出的 PXA255 Xscale 架构的处理机和源码开放的 Linux 嵌入式操作系统的基础上, 实现了视频的实时动态采集、压缩和网络传输, 介绍了系统的研制过程, 给出了系统架构和关键技术。通过构建网络测试平台, 将此编码器接入以太网, 服务器端将接收到的压缩码流进行实时解压、回放。此编码器可用于远程视频监控系统中前端视频的实时采集、压缩和跨 IP 网传输。

关键词: Linux 嵌入式操作系统; PXA255; Xscale 架构的处理机; IP 网络; 视频传输

Design and Application for Embedded Video Codec Based on H.264

FAN Xinnan, XING Chao

(College of Computer & Information Engineering, HoHai Univ., Changzhou 213022)

【Abstract】 This paper presents an embedded video codec that can be applied to industry scene, the video compression standard is H.264. Based on the PXA255 Xscale processor of Intel and the embedded Linux operating system whose source code is opened, dynamic collection, compression and network transmission of video in real time are achieved, the research process of this system is interpreted, the system's framework and pivotal technology are offered. By building network test platform, the codec is connected to Ethernet, and the server decoders and playbacks the received video code flow. Test result shows that this codec can be applied to collection, compression and transmission through IP net in real time of the forward-end video in remote video-supervision system.

【Key words】 Linux embedded operating system; PXA255; Xscale processor; IP Network; Video transmission

基于 IP 网络的嵌入式视频编码器以体积小、功耗低、便携、安装方便等优点, 成为前端视频采集设备的监控系统。尤其值得关注的是此类视频编码器适合特殊行业, 因为其功耗低, 给设备的散热带来了极大的方便。如在安全级别要求很高的矿井工作中, 该编码器作为井下监控终端, 可以作相应的防爆处理, 这是其它设备所无可比拟的。

1 嵌入式视频编码器的硬件构架

使用 400MHz 的 Intel PXA255 处理机, 扩展了所需 SRAM(32MB) FLASH(64kB)、USB 接口、以太网接口、RS232 接口、视频接口采用 USB 接口、电源管理电路, 系统的硬件架构如图 1 所示。

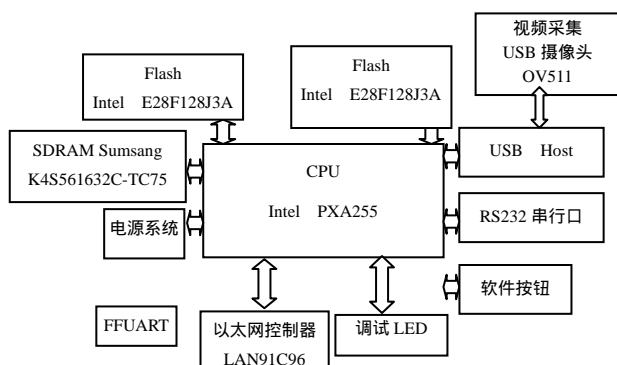


图 1 系统的硬件架构

在此平台上, 成功移植了 Linux 系统, 建立文件系统, 移植了基于 H.264 建议的视频编码库, 以及实现了压缩视频流的网络传输。

2 视频软、硬件接口和驱动实现

嵌入式编码器实现, 首先在硬件平台上成功移植 Linux

并建立交叉编译环境, 其次是视频驱动, 实现应用。

2.1 系统软件流程

系统软件的运行总体流程为: 系统上电后, CPU 读取 BootLoader 程序执行, 初始化后, 进入到 BootMonitor 中, 用户通过设定的启动方式, 下载内核 zImage, 执行启动程序, 最后内核启动完成, 执行用户程序。软件整体流程图 2 所示。

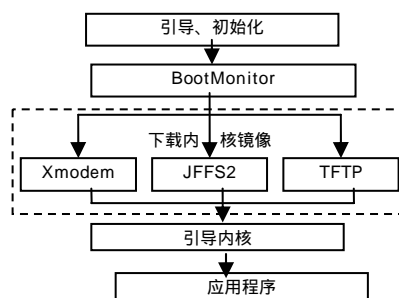


图 2 软件整体流程

2.2 嵌入式 Linux 的视频采集

Linux 对于视频采集设备的支持, 是通过 Video4Linux(V4L)实现的, 现在较高版本的 Linux 内核已包含了 V4L。在编译内核的时候一定要注意以下配置:

- (1)选中编译 Multimedia devices-Video for Linux;
- (2)在 Multimedia devices-Video for Linux->的各项按照视频采集设备的类型进行配置。

Video4Linux 为各种视频卡, 以及并口和 USB 接口的音

作者简介: 范新南(1965—), 男, 副教授、博士生, 主研方向: 图像处理及智能信息系统; 邢 超, 助教、硕士生

收稿日期: 2005-12-07 **E-mail:** fanxn@hhuc.edu.cn

视频采集设备提供统一的编程接口，在 Linux 下，视频采集设备的正常使用依赖于对 Video4Linux 标准的支持。

2.3 嵌入式 Linux 的视频采集驱动开发

Linux 内核提供 Video4Linux 应用程序接口，在程序开发时，首先是基于 Video4Linux API 函数来设计程序。本系统将针对设备文件/dev/video，进行视频捕捉方面的软件开发。

Linux 下视频采集流程如图 3 所示。

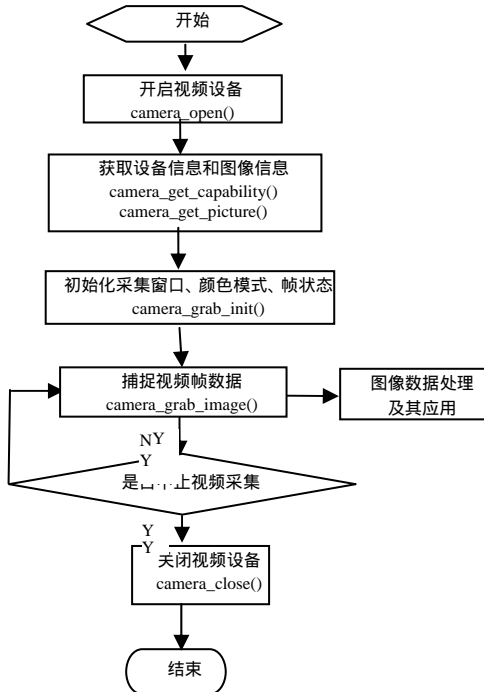


图 3 Linux 下视频采集流程

对应于本视频捕捉设备，可以进行相应的操作，设计这些操作函数时采用了自顶向下的设计方式，规划、定义所用到的函数：

```
int camera_open(char *dev, video_device *vd);
int camera_close(video_device *vd);
int camera_get_capability(video_device *vd);
int camera_get_picture(video_device *vd);
int camera_grab_init(v4l_device *vd,int input,int norm);
unsigned char* camera_grab_image(video_device *vd);
int camera_grab_sync(v4l_device *vd);
int camera_mmap_init(v4l_device *vd);
int camera_get_mbuf(v4l_device *vd);
```

3 嵌入式编码器平台组网及视频传输的实现

嵌入式编码器平台组网测试平台是基于客户机服务器结构，运行于典型的 10Mbps/100Mbps 以太网之下，客户机采用自研嵌入式编码器平台，此类局域网已经十分成熟，图 4 是最简单的结构，当然局域网有更多的结构，它可以更复杂，但对于套接字编程来说，这一切对于我们完全是透明的，因此这个结构已经足以说明问题了。

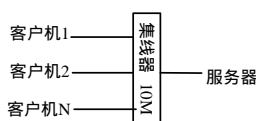


图 4 测试环境

客户机服务器数据交换流程如图 5 所示，该框图和面向流的套接字非常相似。但是具体实现大有不同，此处的套接

字请求连接，接收连接请求，发送数据，数据应答等都是通过自定义的帧类型来完成的。

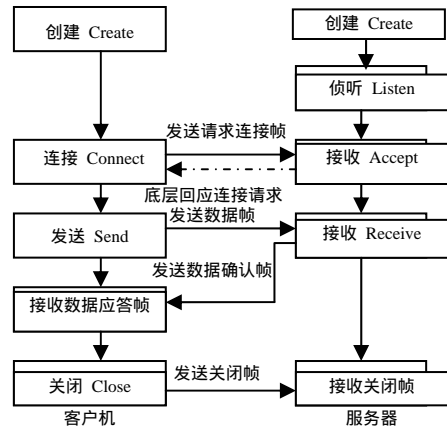


图 5 客户机、服务器通信流程

3.1 套接字的实现方法

本文使用的是面向数据报的套接字，在应用层提供可靠性的保证。即在数据报的基础上再提供了一层封装，包括帧头标志、帧类别、图像格式、滑动窗口大小、帧号、数据长度（有效数据）、数据区。

因此定义了以下几种类型的帧：

```
//帧类别定义
#define frmData 0 //数据帧
#define frmDataAnswer 1 //数据应答帧
#define frmPolling 2 //检测帧
#define frmPlAnswer 3 //检测应答帧
#define frmAbort 4 //异常中止帧
#define frmSysError 5 //系统错误帧
#define frmNormalColse 6 //正常关闭帧
```

并且明确定义了通用帧结构，这个帧结构通用于所有我们支持的图像类型：(1)128*96 格式的图像，每次传输 128B；(2)176*144 格式的图像，每次传输 512B；(3)352*288 格式的图像，每次传输 2 048B；

如缓冲区没有填满，则补零。

这样每种视频格式分别定义最大网络传输单元的目的是为了实时的传输和显示。因为每种格式的视频在单位时间内生成的数据量大小是不同的，而且是成倍数的关系，如果 3 种视频格式的最大传输单元大小相同，相对小的图像，很难填满缓冲区，这样会失去服务端显示的实时性；而对于相对大的图像，由于数据量很大，网络发送的频率很高，容易引起堵塞和缓冲区填满的错误状态。

同时定义了网络套接字的 10 个状态，客户端和服务器端的套接字都在这 10 个状态之间转换。其中初始化状态、接收数据状态，因为其进行的速度很快，终端程序很难显示出来，但这些状态在程序运行过程中是存在的。

```
//网络状态
#define Vedio_STAT_NULL 0 //断开
#define Vedio_STAT_INIT 1 //初始化状态
#define Vedio_STAT_COMM 2 //通信状态
#define Vedio_STAT_STOP 3 //堵塞状态
#define Vedio_STAT_DATA 4 //发送数据状态
#define Vedio_STAT_REC 5 //接收数据状态
#define Vedio_STAT_POLLING 6 //检测状态
#define Vedio_STAT_ABORT 7 //中断状态（没有使用）
#define Vedio_STAT_SYSError 8 //系统错误状态
```

```
#define Vedio_STAT_CLOSE      9 //正常关闭状态
#define Vedio_STAT_WAITFORDATA 10 //等待接收数据
数据帧类型如图 6 所示。
```

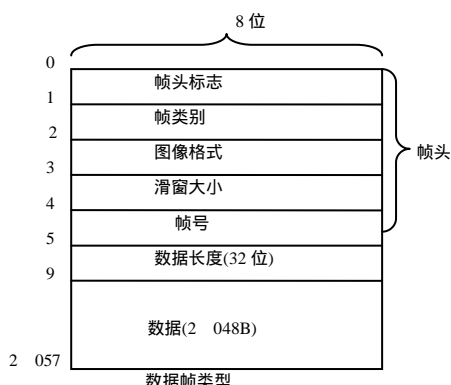


图 6 数据帧类型

3.2 嵌入式编码器（客户端）套接字的定义及实现

在客户端要求客户能主动向服务器发送连接请求；在接收到连接请求应答帧之后，保存对方套接字的地址和端口号，系统处于通信状态；如果有数据，并且系统处于通信状态，先置客户端套接字为发送数据状态，套接字向网络发送数据帧，然后置客户端套接字为堵塞状态，这时缓冲区中即使有数据也不能发送，必须等待对方的应答，这样就避免了大量数据淹没网络的发生；如果，客户端套接字接收到服务端的数据应答帧，客户端套接字恢复到通信状态。

如果在 30s 之内接收不到回应，确定网络中断，关闭本套接字，重新连接。这样客户端套接字处于一个可控的状态转换之中，不会引起系统故障。

3.3 服务器端的套接字的定义及实现

在服务器端，在创建主应用程序时创建侦听套接字，系统处于等待链接状态；当有客户端链接请求到时，侦听套接字创建接收套接字来准备接收数据，如果侦听套接字创建成功，发送检测应答帧，服务器接收套接字处于等待接收数据状态；当有数据到，服务器接收套接字处于接收数据状态，服务端接收套接字验证帧头标志、帧长度、检查帧类型、如果是数据帧就将数据存入缓冲区，向客户端套接字发送数据应答帧，接收套接字处于通信状态；如果服务器接受管理人员的关闭命令，服务器端将发送关闭帧。自动关闭套接字。

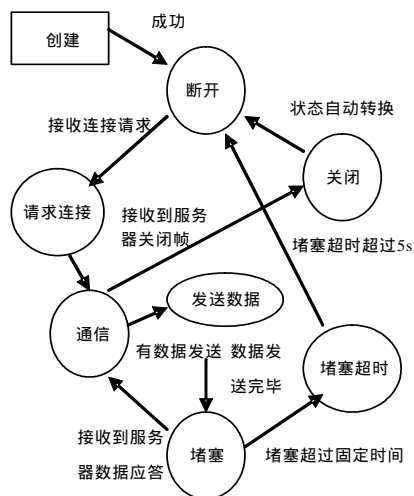


图 7 嵌入式编码器(客户端)套接字状态转换

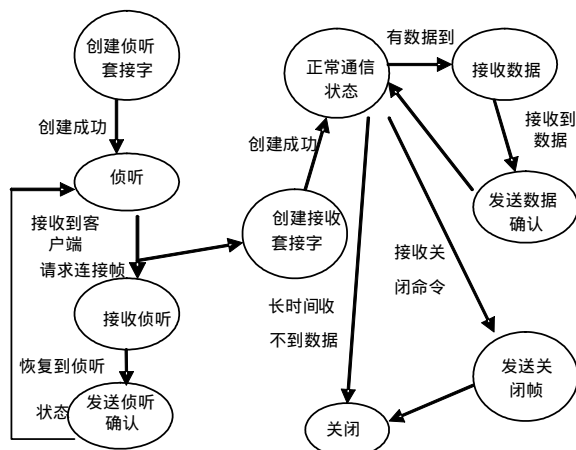


图 8 服务器套接字状态转换

服务端的网络套接字的状态同样处于有限的几个可控的状态之下。通过自定义协议，保证视频传输的可靠性，实现了嵌入式编码器实时采集、压缩和网络传输。

客户端套接字的状态转换，如图 7 所示。服务端网络接收套接字的系统转换，如图 8 所示。

4 总结

本文提出了采用嵌入式技术，在基于 Intel scale 架构的 PXA255 为核心的开发平台上移植了 Linux 操作系统，并且开发了视频驱动软件，将基于 H.264 建议的视频编解码库移植于本平台上，并且通过组网，设计了基于自定义协议的视频网络化传输软件，经过长时间的运行，设计的嵌入式视频编解码器性能可靠，在 QCIF 格式下，可达到 15fps 左右，能够满足监控要求。

系统下一步将从以下几个方面改进：(1)优化编码策略，综合考虑信源信道编码；(2)视频传输策略要改进，提高传输效率，采用解码端留足缓冲区的缓冲机制；(3)采取一些错误弹性机制，提高视频传输的鲁棒性。

参考文献

- 1 李善平, 刘文峰, 王焕龙. Linux 与嵌入式系统[M]. 北京: 清华大学出版社, 2003.
- 2 Hollabaugh C. 嵌入式 Linux[M]. 北京: 电子工业出版社, 2003.
- 3 吴明晖. 基于 ARM 的嵌入式系统开发与应用[M]. 北京: 人民邮电出版社, 2004
- 4 于明俭, 陈向阳, 方 汉. Linux 程序设计权威指南[M]. 北京: 机械工业出版社, 2001.
- 5 丁贵广, 计文平, 郭宝龙. Visual C++ 6.0 数字图像编码[M]. 北京: 机械工业出版社, 2004
- 6 杨 华, 杨松岸, 余松煜. 一种用于 H.264 数据分类的自适应的不平等错误保护策略[J]. 高技术通讯, 2004, 14(3): .11-15.
- 7 ITU H.264. Video Coding for Low Bit Rate Communication[S]. 1998-01.
- 8 高 文. 多媒体数据压缩技术[M]. 北京: 电子工业出版社, 1985.
- 9 何 芸, 翁成坚. 低码率视频压缩编码及全软件 H.264 视频通信系统研究[J]. 中国图像图形学报, 1997, 2(5).
- 10 ITU-T Video coding Expert Group. H.26L Test Model Long Term Number8[Z]. <ftp://standard.pictet.com/video-sit/h26L/tml8.doc>, 2001-07.