

# 实时视频传输系统<sup>①</sup>

陈学涛, 郑力明

(暨南大学 信息科学技术学院, 广州 510632)

**摘要:** 针对智能安防、智慧家庭等实时传输视频的需求, 实现了一种基于 H.264 的嵌入式实时视频传输系统. 系统利用摄像头 TB-2054BD 采集视频信息, 然后经过 DM365 视频协处理器进行压缩, 压缩后的码流采用 RTP 包的形式发送到 PC 客户端进行解码显示. 经验证该系统能够很好的完成实时视频传输的需求.

**关键词:** 智能安防; H.264; DM365; RTP; 视频压缩; 视频传输

## Real-time Video Transmission Platform

CHEN Xue-Tao, ZHENG Li-Ming

(Department of Electronic Engineering, College of Information Science & Technology, Jinan University, Guangzhou 510632, China)

**Abstract:** In order to satisfy the real-time transmission of video required by intelligent safeguard systems and smart home, this paper implements an embedded real-time video transmission platform based on TMS320DM365 processor. This system utilizes the TB-2054BD camera to capture the video, and then send to the co-processor of DM365 to compress the data, the compressed stream will be encapsulated into RTP packets and finally send to the PC client to display after decoded. It has proved that the system has a good performance on real-time video transmission.

**Key words:** intelligent safeguard systems; H.264; DM365; RTP; video compression; video transmission

随着科学技术的发展与人们对安全要求的提高, 银行、交通、商场、智能小区等场合都广泛应用了视频监控系[1]. 嵌入式产品由于具有体积小、管理简单、成本低的特点, 很好地适应了当前安防产品发展的需要; 而基于嵌入式技术的安防产品也成为了视频监控系发展的趋势.

## 1 系统基本原理介绍

### 1.1 DM365 芯片简介

TMS320DM365<sup>[2]</sup>是一款 ARM+DSP 的双核处理芯片, 具有最高 720p@60f/s 的 H.264 硬件编码能力. 其框图如图 1 所示.

视频处理子系统包括视频前端和视频后端, 可以对输入的图像进行前期及后期处理, 增加画质效果; 视频协处理器 HDVICP 可以对高清视频进行 H.264 编解码, 基本不占用 ARM 的资源, 保证了传输系统的

实时性, 也提高了系统的稳定性; 视频协处理器 MJCP 可以对视频或图片采用 MPEG 或 JPEG 进行编码与解码, 从而使 DM365 成为一款真正的多媒体芯片.

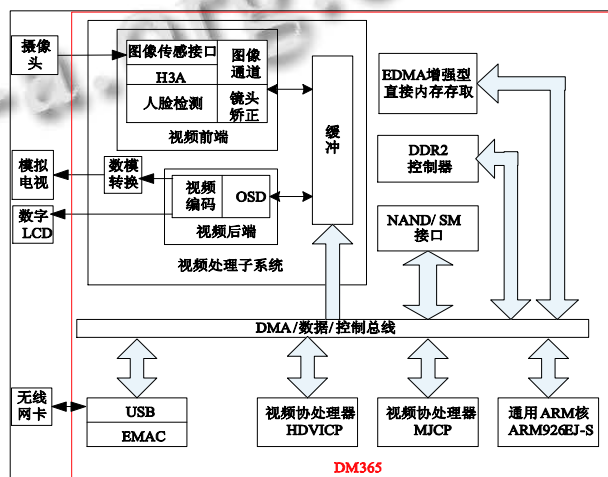


图 1 DM365 芯片框架图

① 基金项目:广东省教育部产学研结合重点项目(2011A090200085)

收稿时间:2013-03-20;收到修改稿时间:2013-04-15

## 1.2 H.264 编码器简介

H.264[3]是国际标准化组织(ISO)和国际电信联盟(ITU)共同提出的继 MPEG4 之后的新一代数字视频压缩格式,其最大的优势是具有很高的数据压缩比率,在同等图像质量的条件下,H.264 的压缩比是 MPEG-2 的 2 倍以上,是 MPEG-4 的 1.5~2 倍. H.264 编码框架<sup>[4,5]</sup>如图 2 所示.

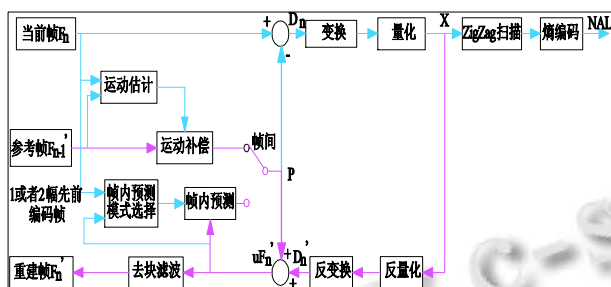


图 2 H.264 编码框架图

编码器的输入是一帧  $F_n$ , 这个帧以宏块(相当于原始图片的  $16 \times 16$  像素)为单位进行处理. 每个宏块可以选择帧内或帧间模式. 在任意一种模式下, 根据重构帧形成预测宏块. 在帧内模式下,  $P$  由前面已经编码、解码和重构的帧  $F'_n$  (图像中的  $\mu F'_n$ ; 意味着  $P$  由未经滤波的样点组成)的样点组成. 在帧间模式下,  $P$  由从一幅或多幅参考帧经过运动补偿后形成的. 在图 2 中, 参考帧是前面已经编码的帧  $\mu F'_{n-1}$ ; 然而, 每个宏块的预测可能由过去或将来已经编码和重构的一帧或两帧(以显示顺序)形成.

残差或差异宏块  $D_n$  是由预测  $P$  宏块减去当前宏块所得. 量化变换系数集  $X$  是由  $D_n$  经过变换(采样块变换)和量化所得. 这些系数被重排序, 然后进行熵编码. 熵编码系数和解码宏块所需要的附带信息(比如宏块预测模式, 量化步长, 描述宏块怎样进行运动补偿的运动矢量信息等)形成压缩比特流. 比特流传向网络抽象层供传输和存储.

## 1.3 RTP 协议简介

RTP<sup>[6]</sup>协议的全称是 Real-Time Transport Protocol, 主要用于多媒体流的实时传输. 通常情况下, 在传输音频、视频数据时都会用到 UDP 的 RTP 协议, RTP 协议为数据流提供时间戳, 实现流同步.

RTP 协议与 UDP 相邻, 并且在 UDP 的上面, 在功能上来说, 它相对于它下面的两层是不相关的, 可是又不

是独立的一层, 一般情况下是用其下的 UDP 协议对采集到的数据进行发送, 从而实现向一个接收点或者多个接收点传输音视频数据. RTP 本身并不是有序的传输而且提供的传输也是不可靠的, 同样也没有对流量、拥塞等制约, 它依靠 RTCP 给它提供帮助.

RTCP 协议的全称是 Real-Time Transport Control Protocol, 和 RTP 配合使用, 用来限制流量和拥塞. RTCP 包中含有已经发送了多少包、丢失了多少包等信息, 服务器可以获得这些数据并且根据这些数据修改传输的速度. 通常情况下 RTP 和 RTCP 要一起用<sup>[7]</sup>, 它们可以以最小的代价获得最好的传输速率, 所以实时音视频数据传输时, 一般采用这两个协议.

## 2 系统架构及关键技术

### 2.1 实验平台搭建

整个系统搭建如图 3 所示.

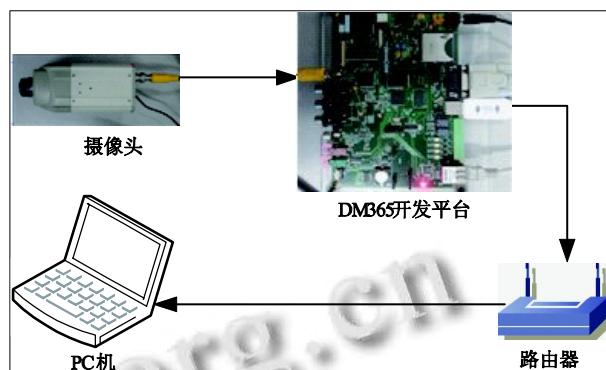


图 3 实验平台

摄像头采集的 PAL 制式复合视频传递给 DM365 开发板, 经过开发板上的视频采样芯片 5150 进行量化转换为 8-bit 的 ITU-R BT.656 格式, 然后送到 DM365 处理器进行视频压缩, 处理完后通过 wifi 发送出去. PC 机通过 wifi 接收开发板发送的数据, 并通过客户端软件进行解码显示.

### 2.2 系统数据流程图

系统可以分为服务器端和客户端<sup>[8]</sup>, 服务器端完成视频采集、压缩及发送, 客户端则完成视频接收、解码和显示. 系统数据流程如图 4 所示, 服务器端通过摄像头视频数据, 然后传递给开发板上视频采样芯片数字化为 YUV4:2:2 格式数据, 接着传递给 DM365 的 HDVIP 压缩芯片进行处理, 处理完码流通过 ARM

端调用 JRTPLIB 库进行封包, 由于要在以太网中进行传输, 故每个数据包长度要小于  $1500-12-8-20=1100$  字节, 其中 12 为 RTP 包头长度, 8 为 UDP 包长度, 20 为 IP 包长度. 客户端界面采用 QT 实现<sup>[9]</sup>, 用于在显示最终解码得到的视频, 其底层调用 JRTPLIB 进行 RTP 包接收, 并通过 FFMEPG 进行解码.

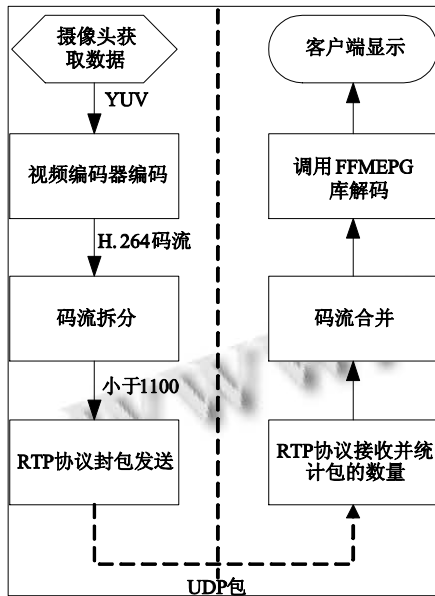


图 4 视频数据流程图

### 2.3 DM365 软件设计

为了使系统具有更好的实时性, 程序设置了一个主线程和 4 个子线程, 如图 5 所示. 主线程用于协调 4 个子线程及管理共享数据. 视频捕获线程采用 V4L2 驱动进行视频数据的获取, 然后放置共享数据区, 供其它子线程调用及处理该数据. 视频显示线程用于将视频捕获线程获取的数据实时显示在 LCD 上. 视频压缩线程用于将视频捕获线程获取的数据压缩成 H.264 码流. 网络发送线程则负责调用 JRTPLIB 库将 H.264 码流打包成为 RTP 包发送出去.

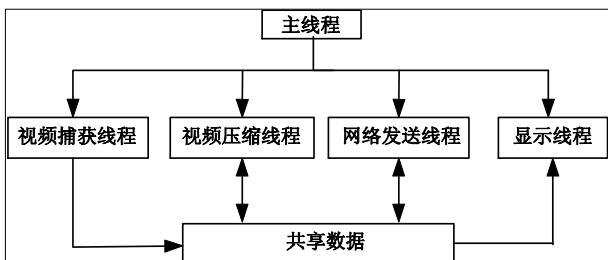


图 5 DM365 软件示意图

### 2.4 wifi 驱动移植

系统传输数据要用到无线传输, 故需要对 wifi 进行移植<sup>[10]</sup>. 实验平台上采用的是雷凌 RT3070, 其移植过程分为以下几步:

- 1) 从雷凌官网下载最新驱动源码.
- 2) 解压后, 进入源码主目录, 修改 Makefile 文件, 添加 DM365 平台支持.
- 3) 修改 os/linux/config.mk 文件, 增加 Wpa\_Supplicant 支持, 用于提供加密功能、TLS 库、配置、控制接口、WPA 提供者等功能.
- 4) make 编译生成内核模块 rt3070sta.ko, 然后拷贝到目标机的根文件系统中, 并将生成的 rt2870.bin 文件放到目标机的/lib/firmware 目录下, RT2870STA.dat 文件拷贝到目标机的/etc/Wireless/RT2870STA.
- 5) 编译内核, 打开 802.11 相关的选项, 使其支持无线网络, 如图 6 所示.

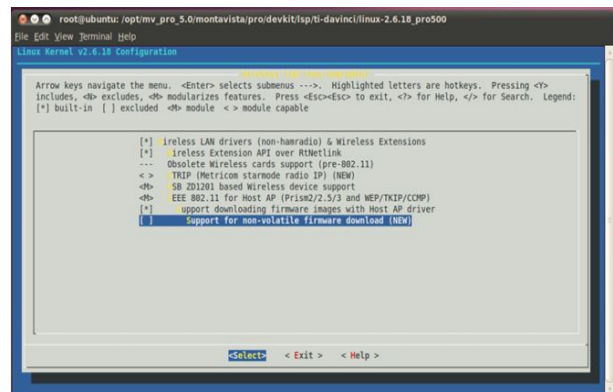


图 6 内核编译选项

- 6) 移植 wpa\_supplicant 工具.
- 7) 移植 wireless\_tools 工具.

### 2.5 JRTPLIB 库移植及程序设计

由于程序要使用 jthread 库提供的线程自动在后台执行对数据的接收, 故需先移植 jthread 库, 下载源码并解压, 然后进行源码目录依次执行下面命令:

```

#./configure --prefix=/opt/nfs/usr/local
--host=arm-montavista-linux CC=arm_v5t_le-gcc
CXX=arm_v5t_le-g++
#make&&make install

```

接着下载 JRTPLIB 库, 并解压, 进入源码目录后, 依次执行下面命令:

```

#./configure --prefix=/opt/nfs/usr/local

```



```
--host=arm-montavista-linux CC=arm_v5t_le-gcc
CXX=arm_v5t_le-g++
#make&&make install
```

JRTPLIB 库<sup>[1]</sup>各模块的功能通过各个类来实现, 如数据的传输、接收存储管理、线程管理等等. 开发过程中常用的类有: RTPSession 类实现了数据接收和发送相关的大部分函数. 如会话创建函数 Create, RTP 分组发送函数 SendPacket 以及游标定位函数 GoToNextSourceWithData 等. 它内置的处理了 RTCP 部分, 不需要额外的进行设计. RTPSessionParams 类主要用来描述 RTPSession 实例将要使用的参数. 相关的成员函数有: 设置时间戳函数 SetOwnTimeStamp Unit, 设置 RTP 分组最大长度函数 SetMaximumPacket Size 等.

RTPUDPV4TransmissionParams 类用于 RTP 发送端口设置, 如函数 SetPortbase, RTCPPacket.

RTCPPacket 类是各种 RTCP 包的基类, 其中两个子类 RTCPPacket 和 RTCPSRPacket 分别用于统计 RTCP 发送报告和接收报告.

JRTPLIB 的接收和发生流程如图 7 所示.

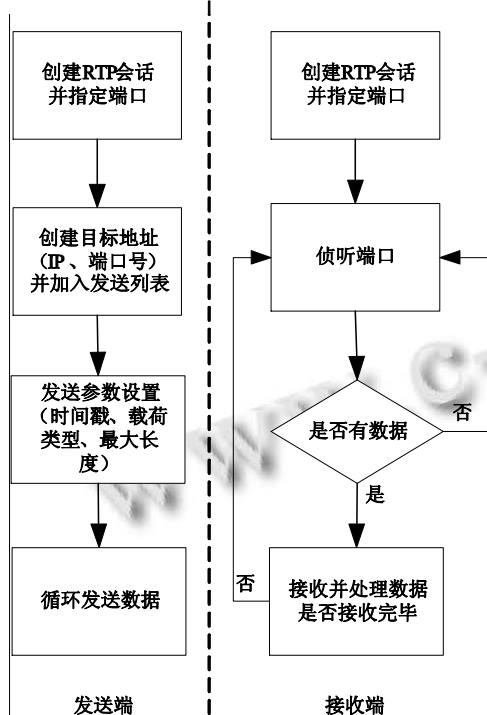


图 7 RTP 库工作流程图

### 3 系统测试及结果

服务器实时显示摄像头捕获的数据如图 8 所示.



图 8 开发板上实时显示

PC 客户端<sup>[1]</sup>解码后显示如图 9 所示, 界面中 Open 按钮为初始化 FFmpeg 解码器, Connect 按钮为初始化 JRTPLIB 接收程序, Receive 按钮使 JRTPLIB 程序进入侦听状态. 经试验, 解码几乎不存在延时, 具有很好的实时性.



图 9 PC 客户端解码显示

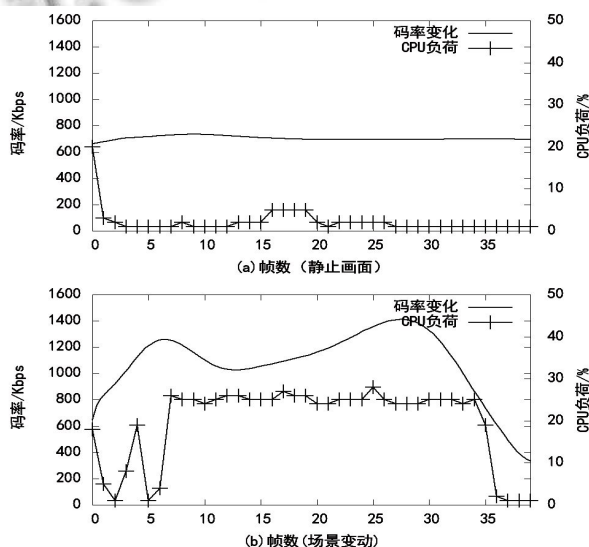


图 10 码率变化

图 10 所示为视频协处理器对摄像头捕获的 PAL 制式的 D1 图像进行 H.264 编码后码率及 ARM 处理器负荷变化情况。

图 10(a)坐标系中上面细曲线为视频协处理器对静止自然场景(固定摄像头)且没有移动物体时进行连续编码时输出码流的速率变化,从图中可以看出码流基本维持在 700Kbps,而且图中下面点画线显示编码过程中对通用处理器 ARM 的占用率也非常低,少于 1%。

图 10(b)坐标系中上面细曲线为视频协处理器对场景不断变动(转动摄像头)的画面进行连续编码时输出码流的速率变化,该曲线有两个峰值,该峰值是由于场景不断变动, H.264 只能采用 I 帧进行编码,故输出码率会相对较高,达到 1.4Mbps。图中下面点画线表明,在场景不断变动的情况下,对通用处理器 ARM 的占用率相对较高,达到 23%左右。当场景再次固定后,码率就会下降,当场景是单一颜色背景时,码率可以降低到 300Kbps 左右。

通过图 10 的两个极端情况,可以得出:当该系统应用于监控时,由于场景不变,偶尔有移动物体,其码率可以维持在 1Mbps 以下,因此可以选用 3G 网络进行视频传输,系统可以适应远程传输视频的需求。

#### 4 结束语

本文提出并实现了一种基于 TMS320DM365 平台的实时视频传输系统,描述了该系统的总体架构和软件的实现方法,并分析了 Dm365 平台是 wifi 驱动移植及 JRTPLIB 库移植和工作原理。实验结果表明,本系统具有良好的实时性和可靠性,能够满足该平台下实时视频传输的需求。

本系统经济适用,本视频传输系统已成功应用于广东省教育部产学研结合重点项目“基于移动自组织

网络的应急通信系统研发与产业化”项目中,并且该系统在远程监控、智能楼宇、智能交通、车载监控等领域具有广泛市场前景。

#### 参考文献

- 1 范亚男,葛卫丽.智能视频监控系统发展及应用.价值工程,2010(6):100-103.
- 2 Texas Instruments Incorporated. TMS320DM36x digital mediasystem on Chip DMSoC video processing front end VPFE user's guide.http://focus.ti.com.cn/cn/lit/ug/sprufg8c/sprufg8c.pdf.2009-03.
- 3 ITU-T Rec. H.264/ISO/IEC11496-10,Advanced Video Coding. Final Committee Draft, Document JVT-E022, 2002-09.
- 4 Iain.E.G.Richardson.H.264/MPEG-4 Part 10 White Paper. http://www.vcodex.com, 2007.
- 5 毕厚杰.新一代视频压缩编码标准.北京:人民邮电出版社,2005:25-65,84-232.
- 6 H.Schulzrinne:a transport protocol for real-time application. http://tools.ietf.org/html/rfc3550.2003-06.
- 7 樊珊.基于 RTP 的 H.264 视频传输技术的研究[博士学位论文].济南:山东大学,2008.
- 8 秦臻,曹剑中.基于 TMS320DM365 的高速网络摄像机的设计.电子设计工程,2011,19(10):121-124.
- 9 倪继利.Qt 及 Linux 操作系统窗口设计.北京:电子工业出版社,2006:40-100.
- 10 孙琼.嵌入式 Linux 应用程序开发详解.北京:人民邮电出版社,2006:30-60,80-100
- 11 Liesenborgs J. The introduction of JRTPLIB.http://research.edm.uhasselt.be/jori/jrtplib/documentation/index.html. 2011-09-08.

(上接第 32 页)

- 2006,(16):121-122.
- 5 余丽君,盖起刚,龙旭梅,刘鹏年.灰色文献——当今不可忽视的重要情报源.现代情报,2007(9):191-193.
- 6 Lynch CA. Institutional repositories: essential infrastrue for scholarship in the digital age. ARL Bimonthly Report, 2003 (226).

- 7 施振宏.论文分类宜用《学科分类与代码》不宜用《中国图书馆分类法》.编辑学报,2005,(5):338-340.
- 8 http://www.baik.com/wiki/%E7%BD%91%E7%BB%9C%E7%88%AC%E8%99%AB.
- 9 何川.科技期刊论文质量的评价指标与方法.科技编辑出版研究文集(第八集),2005,(8):86-89.