

基于达芬奇技术的嵌入式网络摄像机系统研究

肖 军,徐熙平,李沛松

(长春理工大学 光电工程学院,吉林 长春 130022)

摘要: 达芬奇技术是一种专门针对数字视频应用和信号处理的解决方案,它把算法处理和应用层相分离,大大简化了程序编写。介绍了一种基于达芬奇技术的网络摄像机设计方案,详细阐述了该系统的视频采集、H.264 编码实现、RTSP 服务器调度和 RTP 发送。

关键词: 达芬奇技术;H.264;RTSP 服务器;RTP

中图分类号: TP368.1

文献标识码: A

文章编号: 1674-7720(2011)22-0065-03

Research of embedded IP camera system based on DaVinci technology

Xiao Jun, Xu Xiping, Li Peisong

(School of Opto-Electronic Engineering, Changchun University of Science and Technology, Changchun 130022, China)

Abstract: DaVinci technology is a signal processing-based solution tailored for digital video applications that separates the algorithm processing and application layer and greatly simplifies the programming. In this paper, a scheme of IP camera system based on DaVinci is designed, which contains video collection, H.264 encoding, RTSP server scheduling and RTP sending.

Key words: DaVinci technology; H.264; RTSP server; RTP

随着多媒体技术和网络技术的发展,网络摄像机已经广泛运用于网络监控和视频会议领域中。网络摄像机的发展趋势集视频采集、视频压缩、视频发送为一体,并且按照需求添加相应图像识别、图像叠加、运动侦测、云台控制和报警等功能。因此要求处理芯片具有强大的控制能力和数据处理能力。TI 公司推出了达芬奇技术,专门应用于多媒体系统,它把图像处理的编码算法封装成标准 API,应用程序调用这些 API 函数即可实现图像的编码处理。

TMS320DM368 芯片是基于达芬奇技术的音视频数字媒体片上系统(DMSoC)^[1],它以 ARM926EJ-STM 为内核,拥有两个视频图像协处理器(HDVICP 和 MJCP),广泛应用于中低端音视频处理系统。本系统采用 DM368 来构建网络摄像机视频服务器,通过访问服务器可实时查看监控视频。

1 系统总体方案设计

TMS320DM368 只有一个 ARM 内核,通过嵌入 Linux 系统来实现图像的采集处理和网络通信。系统总体框图如图 1 所示,CMOS 图像采集模块把采集的数字图像发送给 TMS320DM368 主芯片,然后调用算法 API 函数实现 H.264 编码。UDP 接收负责接收参数控制命令,经过

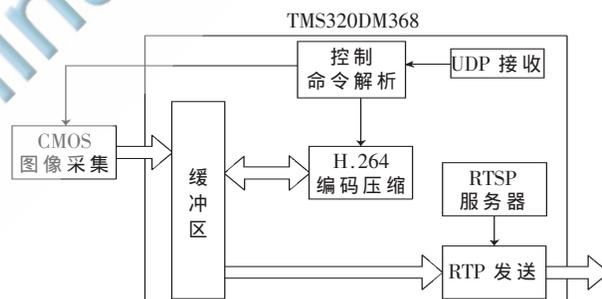


图 1 系统结构框图

命令解析把参数传递给对应模块。RTSP 负责等待客户端访问,然后创建用于 RTP 数据包发送的套接字。

为了实现系统功能,应用程序主要分为以下几个线程:CMOS 视频采集线程、视频编码线程、RTSP 任务调度线程、RTP 视频数据发送线程、UDP 命令接收线程和命令解析线程。CMOS 视频采集线程负责按要求初始化 CMOS 芯片,采集视频图像;视频编码线程调用算法 API 接口,实现数据图像的编码压缩;RTSP 任务调度线程创建 TCP 套接字,等待客户端的访问连接,然后把所访问客户端的地址信息存放在调度任务中;RTP 视频数据发送线程把视频数据通过 RTP 协议发送给指定客户端;UDP 命令接收线程负责接收客户端发送的控制命令,然

技术与方法 Technique and Method

后发送给命令解析线程;命令解析线程把接收到的命令解析处理,把参数传递给对应模块。

2 视频采集和编码

2.1 视频采集模块

视频采集选用 MT9P031 型 CMOS 数字图像传感器芯片,感光面为 1/2.5 英寸,最大像素为 2592×1944。该芯片利用像素跳跃和像素混合技术,在不改变视场的情况下减少输出图像分辨率。它还提供 I²C 接口,主控器可以通过 I²C 接口设置 CMOS 输出图像的属性,并且还集成了 ADC 转换器,直接输出 12 bit 数字图像数据,在时钟和行场同步信号作用下输出每一帧图像,如图 2 所示。每一个场同步信号 FV 有效(高)表示一帧图像,每一个场同步信号 LV 有效(高)表示该帧的一行,在行场同步都有效时每一个像素时钟脉冲的 Dout 输出就是对应像素的 RGB 值。若输出一帧 720 p 的图像,像素大小为 1280×720,每一个 FV 高电平区间就有 720 个 LV 高电平,在每一个 LV 高电平区间又有 1 280 个 PCLK 时钟脉冲。

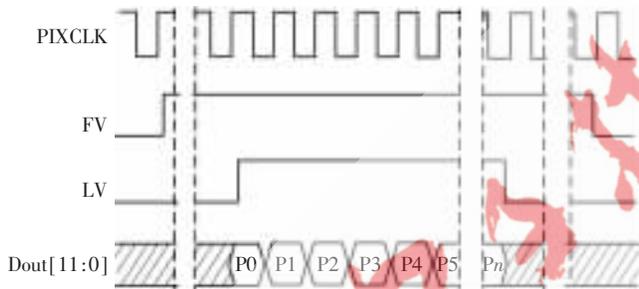


图 2 MT9P031 输出像素时序图

TMS320DM368 有一个 ISIF 接口,专门用于接收数字的视频数据,芯片内部集成了 ISIF 图像接收模块,每接收到一帧完整的数据,芯片就会产生一个 ISIF_INT 中断信号,在中断响应程序里面可以把数据存放到指定地址。MT9P031 和 TMS320DM368 的连接如图 3 所示。MT9P031 芯片的时钟源由 DM368GIO37 引脚输出提供,频率为 24 MHz,GIO91 和 GIO93 引脚分别接复位和使能端。

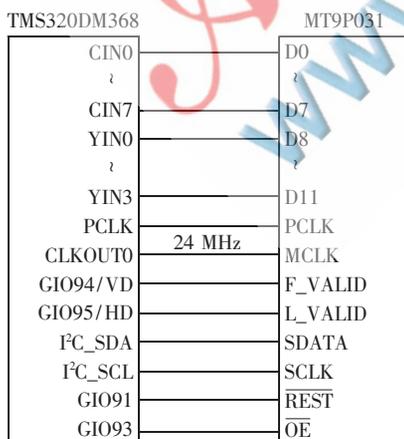


图 3 MT9P031 和 TMS320DM368 连接图

2.2 H.264 编码模块

对所采集的视频数据进行编码是通过调用 Codec

Engine 算法包来实现的^[2]。Codecs Engine 是一组用来调用 xDAIS 算法的 API,不管是在 ARM 端还是在 DSP 端,不管操作系统的差异,调用的 API 函数都是一致的。用户只需要配置相应的 *.cfg 文件即可指定需要调用的算法,然后在应用程序中即可调用 API 函数进行算法编解码。

H.264 是一种高性能的视频编解码标准,继承了 H.263 和 MPEG1/2/4 视频编码器的优点。H.264 标准中定义了两个层次:视频编码层(VCL)和网络抽象层(NAL),VCL 负责高效的数字视频数据压缩,NAL 提供适当的接口对数据进行打包和传送。

为了得到 H.264 格式的视频流,本系统采用的 H.264 算法为 TI 发布的 Codec Engine,配置 alg_server.cfg 文件来实现对已经编译算法的接口声明,文件配置如下:

```
var H264ENC=xdc.useModule
('ti.sdo.codecs.h264enc.ce.H264ENC'); //指定调用的算法
var myEngine=Engine.create("alg_server",[{name: "h264enc",
mod: H264ENC, local: true, groupId: 1}]);
```

其中,name 为调用算法的名字;mod 用来标识实际的算法实现模块;local 指定算法运行在 ARM 端(true)或者 DSP 端(false)。

在应用程序中调用的接口函数有:VIDENC1_create、VIDENC1_control、VIDENC1_process 和 VIDENC1_delete。

3 视频服务器的建立

3.1 搭建 RTSP 服务器

RTSP(Real Time Streaming Protocol)实时流协议是为了建立并控制一个或几个时间同步的连续流媒体^[3],它本身不发送连续媒体流,只充当多媒体服务器的网络远程控制。

RTSP 服务器需要预定义的方法有 OPTION、DESCRIBE、SETUP、PLAY 和 TEARDOWN,在 OPTION 响应中需要实现服务器基本信息描述,DESCRIBE 响应中需要向客户端发送视频的 SDP 信息,SETUP 响应需要根据客户端端口信息建立 RTP(实时传输协议)视频发送套接字,PLAY 响应实现由 RTP 向客户端发送视频数据,TEAREOWN 响应关闭对应 RTP 套接字,并且释放对应的系统资源,流程如图 4 所示。

3.2 RTP 发送 H.264 视频流

RTP(Real-Time Transport Protocol)实时传输协议是在多点传送或单点传送的网络服务上提供端对端的网络传输功能,适用于实时数据传输的场合^[4]。RTP 固定报头格式如图 5 所示。

定义报头的数据格式如下:

```
typedef struct
{
    unsigned char u4CSrcLen:4; //CC
    unsigned char u1Extension:1; //X
    unsigned char u1Padding:1; //P
    unsigned char u2Version:2; //V
```

技术与方法 Technique and Method

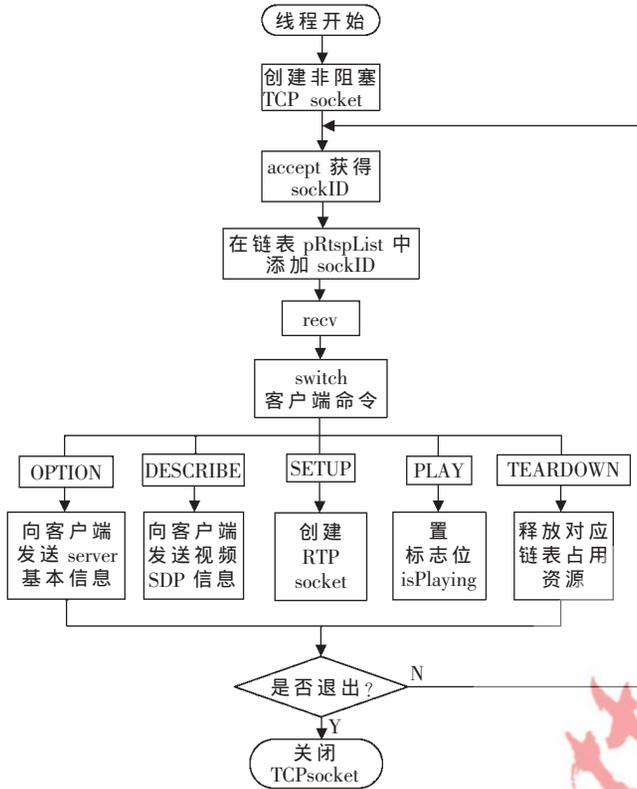


图 4 RTSP 服务器流程图



图 5 RTP 固定报头格式

```

unsigned char u7Payload:7; //PT
unsigned char u1Marker:1; //M
unsigned short u16SeqNum; //SN
unsigned long u32TimeStamp; //时间戳
unsigned long u32SSrc; //SSRC
}RtpHdr;
    
```

经编码模块输出的 H.264 视频流由一系列 NAL 单元 (NALU) 构成, 每个 NALU 由起始码前缀 (0x00000001) 分隔开来^[5], 所以 RTP 发送线程可以在输出编码流中提取出每个 NALU, 经过 RTP 协议转换发送给客户端。从编码流中提取 NALU 单元流程如图 6 所示。

经过验证, 对于 720 p 的 H.264 编码输出 NALU 负载长度不一, 最大达到几十字节, 从网络性能方面考虑, 不宜发送过大的 RTP 数据包, 为此采用 RFC3984 标准的 FU-A 方法对 NALU 进行分割, 设置每个 RTP 包最大长度为 1 400 B。于是对于不同负载长度的 NALU, RTP 发送的数据包格式如表 1 所示。

定义结构体:

```

typedef struct
{
    unsigned char u5Type:5;
    
```

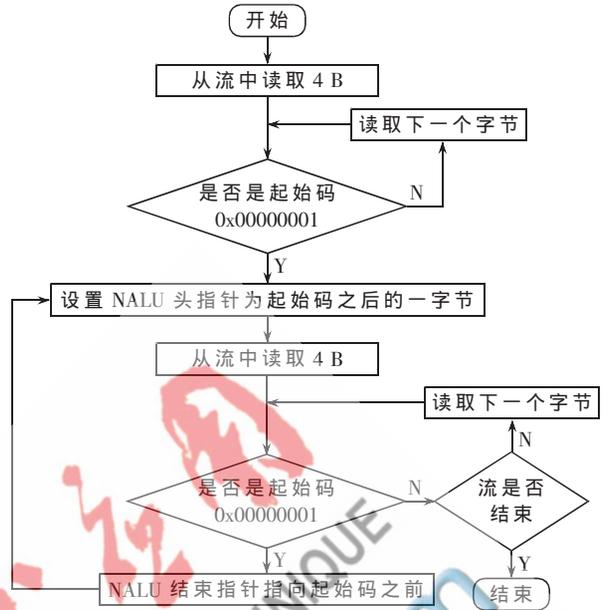


图 6 NAL 单元提取流程图

表 1 RTP 数据发送格式

Nalu 负载 ≤ 1 400	RtpHdr	NaluHdr	NALU 负载
Nalu 负载 < 1 400	RtpHdr	FuIndic	FuHdr NALU 分片负载

```

unsigned char u2Nri:2;
unsigned char u1F:1;
}NaluHdr, FuIndic;
typedef struct
{
    unsigned char u5Type:5;
    unsigned char u1R:1;
    unsigned char u1E:1;
    unsigned char u1S:1;
}FuHdr;
    
```

其中, 参数 NaluHdr 和 FuIndic 由 NALU 第一个字节决定, 参数 FuHdr 用于设置分片发送的起始和结束 RTP 包。

本文研究了基于达芬奇技术的网络摄像机系统, 着重介绍了视频采集、H.264 算法的调用、RTSP 视频服务器以及利用 RTP 数据报发送。系统的应用程序采用多线程技术, 提高了资源利用率和综合性能。搭建的服务器专门用于 H.264 格式的视频流发送, 执行效率高, 便于维护, 对类似系统有很大的参考价值。基于 TMS320DM368 的网络摄像机投入成本低, 开发周期短, 可以广泛运用于各种监控系统。

参考文献

[1] Texas Instruments Incorporated. TMS320DM368 Digital Media System-on-Chip (DMSoC) [EB/OL]. [2011-06-30]. <http://www.ti.com/lit/gpn/tms320dm368>.
 [2] Texas Instruments Incorporated. Codec Engine Application Developer User's Guide [A]. <http://www.ti.com/lit/ug/sprue67d/sprue67d.pdf>.

技术与方法 Technique and Method

- [3] SCHULZRINNE H, RAO A, LANPHIER R. Real time streaming protocol(RTSP)[S]. RFC 2326, 1998.
- [4] SCHULZRINNE H, CASNER S, FREDERICK R. RTP: A transport protocol for real-time applications[S]. RFC 3550, 2003.
- [5] 张多英. 嵌入式视频监控传输系统的设计与实现[J]. 计

算机工程与设计, 2010, 31(4): 724-728.

(收稿日期: 2011-08-22)

作者简介:

肖军, 男, 1985年生, 硕士研究生, 主要研究方向: 嵌入式应用程序开发。

徐熙平, 男, 1969年生, 教授, 博士生导师, 主要研究方向: CCD检测技术相关。

