

# NDK在视频传输系统中的应用

## Application of NDK in Video Transmission System

(清华大学)刘涛 钟晓峰 许希斌

LIU TAO ZHONG XIAOFENG XU XIBIN

**摘要:**DM642 是美国的 TI 公司针对数字视频和图像应用设计的一款高性能数字信号处理器;NDK(Network Developer's Kit)是在 DSP 上进行网络协议开发的工具包。本文研究了 NDK 在 DM642 上的具体应用,并以 MPEG2 编解码库为基础,开发了一个基于 TCP/IP 协议的模块化的网络视频传输系统。此系统软件框架具有结构灵活、易于升级等特点,可应用于各类视频压缩标准的网络。

**关键词:**DM642;NDK;MPEG2;RF5;视频传输

**中图分类号:**TP393.1

**文献标识码:**B

**Abstract:**TI TMS320DM642 is a high-performance processor designed for digital video and image applications. And NDK(Network Developer's Kit)could realize effective TCP/IP stack on DSP. This paper studies the NDK's application development with DM642. And a modularized video transmission system is developed based on MPEG2 codec library. The software architecture of the system is flexible and easy to be upgraded. It's compatible to a variety of video compression standards.

**Key Words:**DM642,NDK,,MPEG2,RF5, Video Transmission.

近年来,信源编码技术有了长足的进步,形成了新一代的技术标准,如 H.264, AC-1 等,为音视频服务提供了新的发展空间。另一方面,TCP/IP 协议逐渐成为视频传输网络的主流。TI 公司针对视频和图像应用推出了数字媒体处理器 DM642。该芯片内部时钟高达 600MHZ,最大处理能力达到 4800MIPS。在音视频方面,集成了 3 个可配置的视频端口(VP),多通道音频串口。在网络接入方面,DM642 集成了 10/100Mbps 的以太网 MAC,适用于开发以太网多媒体通信设备。本文通过开发一个基于 TCP/IP 的模块化的网络视频传输系统,重点讨论了 NDK 在 DM642 上的应用问题。

## 1 NDK 和 RF5 简介

TI 的集成开发环境中的实时操作系统是 DSP/BIOS, 它并不包含网络功能。为了解决这个问题,TI 结合其 C6000 系列推出了 NDK(Network Developer's Kit),该开发包可以用较少的资源消耗支持 TCP/IP。NDK 仅用 200~250kB 程序空间和 95kB 数据空间即可支持常规的 TCP/IP 服务,包括应用层的 telnet, DHCP, HTTP 等。所以,NDK 很适合目前嵌入式系统的硬件环境,是实现 DSP 上网的重要支撑工具。

软件参考框架(Reference Framework)是 TI 为使用 DSP/BIOS 和 TMS320DSP 算法标准的应用开发者提供的。参考框架有不同的版本,其中 RF5 适合在 C6000 系列 DSP 上使用多任务、多通道和多算法来开发较复杂的应用。RF5 包括 4 种元素:任务(Task)、通道(Channel)、单元(Cell)和算法(Algorithm)。算法是最基本的元素,每一个算法完成一项特定的工作,然后按 XDIAS 算法规范封装成一个单元,一个或多个单元注册进通道后,在通道里按一定的顺序执行。一个任务中可以放入一个或多个通

道,以实现特定的应用。下面就用一个具体的实例说明如何在 DM642 上利用 NDK 实现一个简易的视频传输系统。硬件平台是 TI 的 DM642EVM,软件环境是 CCS 2.20.18。

## 2 视频传输系统的软件架构

TI 公司为 DM642EVM 提供了例程 mpeg2\_loopback, 集成了 MPEG2 编码和解码库。它使用 RF5,运行三个任务:输入任务,处理任务和输出任务。它们的优先级是相同的,为避免发生资源抢占的情况,任务之间用 SCOM 进行同步和通信。输入任务从 NTSC 输入设备中获取帧,转换成 YUV4:2:0 格式,然后向处理任务发 SCOM\_message,在 message 中包含帧的地址;处理任务把帧进行 MPEG2 编码,然后把比特流传给解码通道进行解码,再传给输出任务;输出任务将帧转换成 YUV4:2:2 的格式后在 NTSC 输出设备上显示,然后向输入任务发一个 SCOM\_message,告诉它可以继续采集下一帧。三个任务用 SCOM\_message 构成一个回环(loopback)。如图 1 所示。

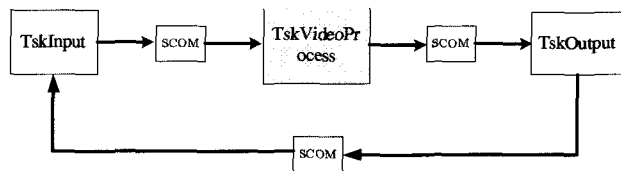


图 1 三个 Task 用 SCOM 组成 loopback

## 3 网络传输模块开发

上述例程把编码和解码都放在 TskVideoProcess 中,通常用来演示。实际应用中编码和解码往往在不同的地方,比如视频监控,一端采集图像、编码、通过以太网或其他方式发出去,另一端接收下来进行解码,输出显示。因此有必要把编码和解码分开,然后添加网络。

刘涛:硕士研究生

### 3.1 将编码和解码分开

把原来的“处理任务”分成“编码任务”和“解码任务”。输入任务和编码任务用一对 SCOM 消息通信。输入完成后发一个 SCOM 消息给编码任务,消息中包含帧的指针。编码任务开始工作,完成后再次通知输入模块。由于编码前的帧和编码以后的帧放在不同的内存区域,这样就保证不会出现内存读写冲突的情况。同理,解码模块和输出模块也利用一对 SCOM 消息通信。如图 2 所示。

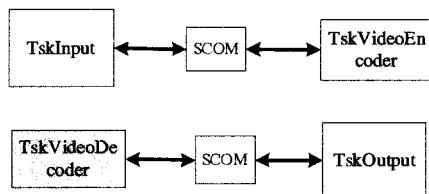


图2 打开原来的闭环,组成新的同步方式

下面截取编码任务 tskEncode.c 中的代码来说明 SCOM 的用法。

```
ScmBufChannels *pMsgBuf;
ScmBufChannels scmBufMessage;
while (1) {
//等待输入任务的消息,如果没来则一直等下去
pMsgBuf = SCOM_getMsg (fromInputtoEnc, SYS_FOREVER);
..... //编码
//编码完毕,将 SCOM 还给输入任务,使其继续采集视频
SCOM_putMsg (fromEncToInput, pMsgBuf);
//编好的数据流存在 bitBuf 里面,将这个地址放在 SCOM 中
scmBufMessage.bufChannel = bitBuf;
//向网络发送模块发消息,消息中含有编好的文件地址
SCOM_putMsg (fromEncToNet, &scmBufMessage);
//等待网络发送模块送回 SCOM,如果没送回就一直等下去
SCOM_getMsg (fromNettoEnc, SYS_FOREVER);
}
```

### 3.2 添加网络发送任务和接收任务

在 BIOS 里创建一个静态的任务,取名 network\_main。在 network\_main()里打开系统,配置好 IP 和子网掩码,然后创建一个动态的任务,完成数据的发送或接收。假设发送函数名为 taskNetworkTx,任务优先级是 5,创建的语句是:hSend = TaskCreate (tskNetworkTx, "NetTx", 5, 0x2000, 0, 0, 0);接收端为 tskNetworkRx,创建的语句是:hRecv = TaskCreate (tskNetworkRx, "NetRx", 5, 0x2000, 0, 0, 0);tskNetworkTx 和 tskNetworkRx 两个函数内部是具体的套接口(socket)编程,根据实际需要可以采用 TCP 或 UDP 协议。以发送端为例,采用 UDP 协议,代码如下:

```
void tskNetworkTx () {
.....//声明 socket,打开 SCOM
```

//打开任务

```
fdOpenSession (TaskSelf ());
//sudp 是数据报套接口
sudp = socket (AF_INET, SOCK_DGRAM, IPPROTO_UDP);
//配置目标 IP 地址和端口号
bzero (&sinl, sizeof (struct sockaddr_in));
sinl.sin_family = AF_INET;
sinl.sin_len = sizeof (sinl);
```

```
sinl.sin_port = htons (7);
sinl.sin_addr.s_addr = inet_addr ("192.168.0.7");
for (;){
//等待编码器将数据指针发送过来
pMsgBuf = SCOM_getMsg (EncToNet, SYS_FOREVER);
//将数据指针存到 mpg_buf 里
mpg_buf = pMsgBuf->bufChannel;
//发送以 mpg_buf 开始的 1KB 数据
sendto (sudp, (UINT8 *) mpg_buf, 1000, 0, &sinl, sizeof
(sinl));
//将 SCOM 还给编码器
SCOM_putMsg (NettoEnc, pMsgBuf);
}
}
```

添加网络收发模块后的系统流程如图 3 所示。

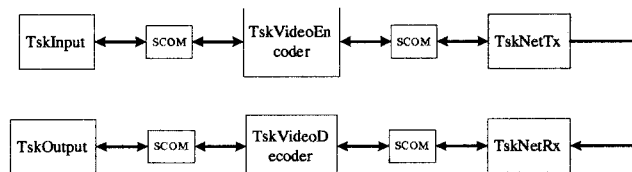


图3 添加网络收发模块后的流程图

## 4 总结与展望

本视频传输系统在 DM642EVM 上调试、运行通过,实现了 MPEG2 实时视频压缩/解压缩和以太网传输等功能。当前在以 DSP 为硬件环境的系统中应用 TCP/IP 协议已成为开发热点。随着新一代视频压缩标准的普及,以及以数字机顶盒为代表的网络流媒体的广泛应用,集成了网络模块的视频处理芯片必将拥有更广阔的应用前景。

本文作者创新点:以一个具体实例——MPEG2 视频传输系统说明了 NDK 的应用过程以及需要注意的问题,列出了部分重要的代码,并做了详细的说明,对在 TI DSP 上进行网络模块的开发具有重要的借鉴意义。

参考文献:

- [1]TI TMS320C6000 TCP/IP Network Developer's Kit User's Guide
- [2]TI Reference Frameworks for eXpressDSP Software: RF5, An Extensive, High-Density System
- [3]TI TMS320C6000 TCP/IP Network Developer's Kit Programmer's Reference Guide
- [4]刘喜龙,石中锁 基于 H.264 的嵌入式视频服务器的设计[J]微计算机信息,2005,21(1):133-134
- [5]杜文,沈勇,唐昆 DM642 上 TCP/IP 协议的实现及性能测试[J].微计算机信息,2006,22(2):149-152

作者简介:刘涛(1982.8-),男,清华大学硕士研究生,专业:信息与通信工程;钟晓峰(1977.8-),男,清华信息科学与技术国家实验室(筹)助理研究员;许希斌(1968.3-),男,清华大学信息技术研究院研究员。

**Biography:**LiuTao (1982.8-),male,master candidate of Department of Electronic Engineering, Tsinghua University.

(100083 北京 清华大学 电子工程系)刘涛 钟晓峰 许希斌  
通讯地址:(100083 北京 清华大学 电子工程系)刘涛

(收稿日期:2007.1.23)(修稿日期:2007.2.25)