基于DSP的嵌入式TCP/IP协议栈在μC/OS－II上的实现

黄林生，林 岩
（北京航空航天大学 自动化科学与电气工程学院，北京 100083）

摘 要：随着以太网络技术和数字信号处理技术的飞速发展，如何使嵌入式处理器具备网络通讯的功能成为目前研究的新领域。结合以DSP TMS320C6713为主处理器的SEED－DEC6713嵌入式控制板和新型10/100M自适应以太网控制器LAN9218I，主要讨论了如何扩展SEED－DEC6713的网络通讯模块，如何编写网卡的驱动程序，以及如何在嵌入式实时操作系统μC/OS－II上移植TCP/IP协议栈LwIP的问题。经过测试将所开发的这套硬件系统应用于网络音频数据的监听，收到了实效。

关键词：TMS320C6713；LAN9218I；网络模块；LwIP；μC/OS－II
中图分类号：TP316.84 **文献标识码**：A **文章编号**：1673－629X(2008)06－0195－04

Realization of Embedded TCP/IP Stack on μC/OS －II Based on DSP

HUANG Lin-sheng, LIN Yan
(School of Automation Science and Electrical Engineering of Beihang University, Beijing 100083, China)

Abstract: With the fast development of the Ethernet technology and digital signal process technology, at present how to have the MPU possess the function of network communication is becoming a new realm. Combining SEED－DEC6713 which CPU is DSP TMS320C6713 and new pattern 10/100M auto－adapted Ethernet controller LAN9218I, the problems of how to extend the network communication module of SEED－DEC6713, how to compile the driver of net card, how to transplant TCP/IP stack LwIP based on embedded real－time operating system μC/OS －II are mainly discussed in this paper. After tested, the hardware system which is designed in this paper is applied to the monitor of network audio－data, which receives actual effect.

Key words: TMS320C6713; LAN9218I; network module; LwIP; μC/OS －II

0 引 言

随着嵌入式设备与网络的日益结合，在嵌入式实时操作系统中引入TCP/IP协议栈，以支持嵌入式设备接入网络，成为嵌入式领域的一个重要研究方向，嵌入式设备由于其体积小、成本低、开发方便等优点已经在民用和工业领域得到广泛的应用。在笔者所在单位研制的SEED－DEC6713嵌入式硬件系统的基础上，主要讨论了如何扩展硬件系统的网络通讯功能，如何结合该硬件系统平台，将被广泛应用的轻型LwIP协议栈在μC/OS －II系统上移植实现及网卡驱动程序的编写，最终实现系统的网络通讯功能。将文中设计的这套系统应用于网络音频数据的监听，经过测试收到了实效。

收稿日期：2007－09－30
基金项目：国家自然科学基金资助项目(60174001)
作者简介：黄林生(1982－)，男，硕士研究生，研究方向为控制工程与网络通信；林 岩，博士生导师，研究方向为鲁棒控制与自适应控制。

1 硬件平台及开发设计

1.1 硬件平台

本设计采用的硬件平台是笔者所在单位研制的SEED－DEC6713，它是SEED系列嵌入式DSP控制模板中的一员，其上边包括了：高性能的32－位浮点DSP：TMS320C6713，主频可达300MHz，处理速度高达2400MIPS/1800 MFLOPS，C6713采用2级Cache结构，片上共有264k×8－位存储器（其中4k×8－位L1P Cache、4k×8－位L1D Cache、256k×8－位L2 RAM /Cache）。并外扩了高速、大容量存储器（100MHz的SBSRAM、SDRAM）；具有一路标准的Line In输入，1路标准的Microphone输入，1路标准的Audio Line Out输出；两路接口标准可切换的RS232/RS422/RS485通用异步接口；标准的USB2.0控制模块；IIC总线串行EEPROM；标准的EMIF扩展总线，

外设扩展端口，HPI 扩展端口，McASP 扩展端口。除此之外，控制板上配有 ESAM 加密模块，可以更好保护开发者的知识产权[1]。

1.2 对 SEED－DEC6713 的网络扩展

为了使 SEED－DEC6713 能够与以太网络通信，必须对它进行网络模块的扩展。网络模块处在网络协议的最底层－物理层，它是实现网络通讯的基础。本系统采用的是 SMSC(美国史恩希)公司的 LAN9218I 型网络控制器，之所以选择 LAN9218I，首先因为它是 10M/100M 的自适应以太网控制器，可以满足现代高速的以太网传输要求，而且它是专门为嵌入式设备设计的，可以很方便与嵌入式处理器实现接口设计，这可以加快开发工作的进度。其次就是它具有大容量的 FIFO 16－kbyte 能够容纳多达 200 个数据包，内部具有专门存储 MAC 地址的寄存器，这样可以省去外部 EEPROM 的连接，避免了 MAC 地址从外部的读写操作，降低了开发的难度，同时也降低了开发的成本[2]。

DSP TMS320C6713 和 LAN9218I 的接口电路设计如图 1 所示。

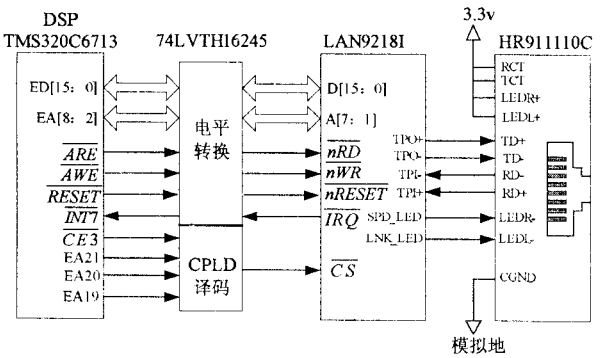

图 1 TMS320C6713 和 LAN9218I 硬件连接原理框图

2 LwIP 协议栈

LwIP 是一套用于嵌入式操作系统、开放源代码的 TCP/IP 协议栈。LwIP 可以移植到操作系统上，也可以在无操作系统的情况下独立运行。LwIP 协议栈的实现是在保持 TCP/IP 协议栈基本功能的基础上，减少对硬件资源的占用。一般它只需要几十 k 的 RAM 和 40k 的 ROM 就可以运行了，这使 LwIP 协议栈非常适合在低端的嵌入式系统中使用。LwIP 的特性包括：支持多网络接口下的 IP 转发；支持 ICMP 协议；支持扩展性的 UDP 报文；支持转发的 TCP 报文；提供专门的内部回调接口(Raw API)用以提高应用程序性能等；LwIP 使用的进程模型是所有协议都驻留在一个单独的进程中，与操作系统内核是分离的；应用程序可以驻留在 LwIP 进程，也可以是分离进程；TCP/IP 和应用程序之间的通讯，可以通过函数调用(应用程序与 LwIP 共享进程时)，也可以通过更抽象的 API。这种与操作系统分开的进程模型可以方便在不同的操作系统上进行移植，LwIP 的进程模型如图 2 所示[3,4]。

应用层	FTP	HTTP	Tenlet
传输层	TCP		UDP
网络层	ICMP		IP
网络接口层	ARP Packet		IP Packet

图 2 LwIP 进程模型

3 LwIP 在 μC/OS－II 下的移植

μC/OS－II 是一种公开源代码的、占先式多任务的微内核 RTOS，它具有以下几个优点：源代码公开、代码结构清晰、注释详尽、可移植性好、可以方便裁减、最多可以开启 64 个任务。μC/OS－II 采用了基于优先级的占先式实时内核，它包含了实时内核、任务管理、时间管理、任务间的同步(信号量、邮箱、消息队列)和内存管理等功能。但是 μC/OS－II 始终只是一个实时的调度和通讯内核，缺少对外围设备和接口的支持，为了使其能够与以太网络通讯，就必须对其进行 TCP/IP 协议栈的移植[5]，文中采用轻型 TCP/IP 协议栈 LwIP。

为了实现 LwIP 在操作系统 μC/OS－II 上的移植，只需要实现 LwIP 的操作系统模拟层即可，在操作系统模拟层实现了定时器、处理同步、消息传送等与操作系统相关的功能，具体操作系统模拟层的实现主要分为如下的几个部分。

3.1 与 CPU 或者编译器相关部分

在 cc.h，cpu.h，perf.h 这些头文件里面主要定义高低位的存储顺序和字节对齐问题，由于 μC/OS－II 是在 DSP TMS320C6713 上移植的，所以必须有以下的定义：

```
#define BYTE_ORDER LITTLE_ENDIAN //定义为小端存储模式
typedef unsigned char u8_t;
typedef signed char s8_t;
typedef unsigned short u16_t;
typedef signed short s16_t;
typedef unsigned int u32_t;
typedef signed int s32_t;
```

C 语言的结构体一般是 4 字节对齐的，在数据包进行处理时，LwIP 是根据数据结构体中数据的不同数据的长度来读取相应数据的，所以这里面就需要使用

关键字_packet,这样可以让编译器放弃struct的字节对齐,所以LwIP中定义了几个PACKED_FIELD_xxx宏,

```
#define PACK_STRUCT_FIELD(x) x
#define PACK_STRUCT_STRUCT
#define PACK_STRUCT_BEGIN
#define PACK_STRUCT_END
```

3.2 与操作系统相关的部分

sys_arch.h和sys_arch.c中的内容是与操作系统相关的一些结构和函数,主要实现了信号量、消息、超时、线程这四部分:

1) sys_sem_t信号量。

```
struct sys_sem_t
sys_sem_new() //创建一个信号量结构
sys_ sem free() //释放一个信号量结构
sys_ sem signal() //发送信号量
sys_ arch_sem wait() //请求信号量
```

2) sys_mbox_t消息。

```
sys_mbox_new() //创建一个消息队列
sys_mbox_free() //释放一个消息队列
sys_mbox_post() //向消息队列发送消息
sys_arch_mbox_fetch() //从消息队列中获取消息
```

3) sys_arch_timeout函数。

LwIP中每个与外界网络连接的线程都有自己的timeout属性,即等待超时时间。这个属性表现为每个线程都对应一个sys_timeout结构体队列,包括这个线程的timeout时间长度,以及超时后应调用的timeout函数,该函数会做一些释放连接,回收资源的工作。如果一个线程对应的sys_timeout为空(NULL),说明该线程对连接做永久的等待。

timeout结构体已经由LwIP自己在sys.h中定义好了,而且对结构体队列的数据操作也由LwIP负责,所要实现的是如下函数:

```
struct sys_timeouts * sys_arch_timeouts(void)
```

这个函数的功能是返回目前正处于运行态的线程所对应的timeout队列指针。timeout队列属于线程的属性,因此是OS相关的函数,只能由用户实现[6]。

4) sys_thread_new创建新线程。

LwIP既可以单线程运行也可以多线程运行,单线程情况下只有一个tcpip线程(tcpip_thread),负责处理所有的tcp/udp连接,各种网络程序都通过tcpip线程与网络通讯。在多线程情况下运行,可以提高效率,降低编程复杂度。但是必须实现创建新线程的函数:

```
void sys_ thread_ new(void ( * thread)(void * arg), void * arg);
```

由于μC/OS-II系统中没有线程(thread)的概念,只有任务(Task)。所以这时只需要把系统的创建任务函数OSTaskCreate封装一下,就可以实现sys_thread_new。但是由于LwIP中thread没有像μC/OS II中优先级的概念,所以实现时要由用户事先为LwIP中创建的线程分配好优先级,这里是通过宏定义为用户线程分配好优先级的[7,8]。

3.3 lib_arch中库函数的实现

由于机器的数据格式和网络数据的格式存在着差异,在从网络接收到数据或者从机器向网络发送数据都存在一个数据格式的转换问题,所以在LwIP协议栈中要实现以下四个外部函数:

```
u16_t htons(u16_t n); //16位数据由机器数据格式到网络数据的格式的转变
u16_t ntohs(u16_t n); //16位数据由网络数据格式到机器数据的格式的转变
u32_t htonl(u32_t n); //32位数据由机器数据格式到网络数据的格式的转变
u32_t ntohl(u32_t n); //32位数据由网络数据格式到机器数据的格式的转变
```

3.4 网络设备驱动程序的实现

本设计中采用的网络芯片是LAN9218I,这一款专门为嵌入式设备准备以太网控制器,目前在国内市场开发度基本为零,所以笔者根据LwIP的驱动程序的编写框架自己动手开发了LAN9218I的驱动程序,放在.\arch\netif\9218netif.c中,希望对后续开发能起到应有的作用。

在LwIP中可以有多个网络接口,每个网络接口都对应了一个struct netif,这个netif包含了相应网络接口的属性、收发函数。LwIP调用netif的方法netif->input()及netif->output()进行以太网packet的收、发等操作。在驱动中主要做的,就是实现网络接口的收、发、初始化以及中断处理函数。驱动程序工作在IP协议模型的网络接口层,它提供给上层(IP层)的接口函数如下:

```
//网卡初始化函数,调用网卡芯片初始化 low_level_init(struct netif * netif)
void Lan9218_init(struct netif *netif),
//网卡接收函数,从网络接口接收以太网数据包并把其中的IP报文向IP层发送
//在中断方式下由网卡ISR调用,调用底层接收函数 struct pbuf * low_level_input(struct netif *netif)
```

void Lan9218_input(struct netif * netif)

//网卡发送函数,给 IP 层传过来的 IP 报文加上以太网包头并通过网络接口发送,调用底层发送函数 err_t low_level_output(struct netif * netif,struct pbuf * p)

err_t Lan9218_output(struct netif * netif, struct pbuf * p, struct ip_addr * ipaddr)

//网卡中断处理函数 ISR,用于数据来临时数据的接收工作

void Lan9218_isr(void)

以上的函数都可以分为协议栈本身的处理和对网络接口硬件的操作两部份,对硬件的操作是对上层屏蔽的。

4 系统的调试程序

在所有的移植工作结束以后,就可以建立 main 程序了,在 main 函数里面要初始化 μC/OS-II,创建一个主任务 Task_lwip_init 任务优先级设为最高(0),并且在该主任务里面初始化 LwIP,还创建了 tcpip_thread(优先级 5)和 tcpecho_thread(优先级 6)两个线程,其中 tcpip_thread 线程是主线程,tcpecho_thread 是用 LwIP API 函数实现的一个监听 7000 端口的服务器,它可以接收来自远程主机的客户请求,并可以与客户机之间进行数据的收发操作。这里有一点需要注意的是 LwIP 的初始化必须在 μC/OS-II 完全启动之后也就是在任务中进行,因为它的初始化用到了信号量等 OS 相关的操作。main 函数关键代码如下:

```
main()
{
    …… //中断、板卡的初始化
    OSInit(); //uC/OS-II init
    OSTaskCreate(Task_lwip_init,(void *)0,&Task_lwip_init_stk[4096-1],0); //main task
    OSStart(); //enter Idle model,wait for another higher priority task
}
```

由于 TCP/IP 协议栈代码量比较大,处理数据任务非常繁琐,所以文中利用了 TMS320C6713 的二级缓存技术。用于提高了数据处理的速度。

编译运行后,在 PC 机上向板卡发送数据,建立连接后,数据可以返回 PC 机。用 ping ip 地址命令可以得到 ICMP reply 响应。这说明 ARP、ICMP、IP、TCP 协议都已正确运行。

5 结束语

这套系统应用于音频数据的网络监听,即通过板卡上 AIC23 现场采集语音数据(可以是电话、广播、电台等),将模拟语音数据转化为数字语音,然后通过 TMS320C6713 进行数据压缩,通过网络传输到远程终端(PC 机等)。在经过对 SEED-DEC6713 系统板的进行网络模块扩展以后,移植了实时操作系统 μC/OS-II,解决了采集、压缩、传输各个任务之间的系统调度问题;又在此基础上移植了嵌入式 TCP/IP 协议栈 LwIP,实现 SEED-DEC6713 的网络通讯功能,解决了系统板数据的网络传输问题。经过测试该系统板可以满足网络实时性、稳定性的要求,并且具有硬件少、结构简单、容易扩充等优点。

参考文献:

[1] TMS320C6713 DataSheet[R]. [s.l.]: Texas Instruments, 2001.

[2] LAN9218I DataSheet[EB/OL]. 2005. http//www.smsc.com.

[3] EComer D. 用 TCP/IP 进行网际互连(第 1 卷):原理[M]. 北京:电子工业出版社,2001.

[4] Dunkels A. LwIP source code[DB/OL]. 2002. http://prdownloads.sourceforge.net/ucos-lwip-c6x/ucos-lwip-c6x-src-1.0.0.

[5] Labrosse J. μC/OS-II 源代码公开的实时嵌入式操作系统[M]. 北京:中国电力出版社,2001.

[6] 杨天怡,陈 禾,柴 毅. μC/OS-Ⅱ支持下的嵌入式 TCP/IP 协议应用[J]. 计算机科学,2006,33(3):72-74.

[7] 伊文斌,周贤娟,鄢化彪,等. μIP TCP/IP 协议分析及其在嵌入式系统中的应用[J]. 计算机技术与发展,2007,17(9):240-243.

[8] 朱华均. UC/OS-Ⅱ操作系统在 ARM 处理器上的移植[J]. 计算机工程,2004(S1):64-65.

(上接第 194 页)

(PHY) Extension in the 2.4 Ghz Band[S]. [s.l.]: IEEE, 1999.

[2] Samsung Electronics. S3C2410X 32-Bit RISC Microprocessor, User's Manual[M]. [s.l.]: Samsung Electronics, 2003.

[3] Malinen J. Host AP Linux driver[EB/OL]. 2005. http://hostap.epitest.fi.

[4] The linux-wlan Company. The linux-wlan Project[EB/OL]. 2005. http://www.linux-wlan.org.

[5] 王 恒,吴宝明,林金朝. 基于 ARM 和 CDMA-1X(GpsOne)的移动远程心电监护终端的设计[J]. 医疗卫生装备,2006,27(1):7-9.