

Telnet 在 8 位机上的实现

叶勇杰¹, 高仕斌¹, 黄建敏²

(1.西南交通大学电气工程学院, 四川成都 610031; 2.西南交通大学峨眉校区电气系, 四川峨眉 614000)

摘要: 嵌入式系统与以太网互联是近年来的一个热点。由于 16 位和 32 位机的资源相对丰富, 上网方案相对成熟, 都是基于大内存容量的协议栈。但是, 在 8 位低端处理器上网方案中, 要移植这些协议栈就不太现实。本文讨论了 uIP 协议栈在 8 位机上的应用, 并且成功建立了一个稳定的网络 Telnet 服务器, 实现了对系统的远程监控。

关键词: 嵌入式 Internet; uIP; TCP/IP 协议栈; C8051F

中图分类号: TP393.02 文献标识码: B 文章编号: 1009-9492 (2006) 12-0041-04

1 引言

在 8 位处理器上实现以太网连接, 必须考虑到它的不利条件: 内存空间较 16 位以上处理器小, 资源紧缺。因此, 在其上实现 TCP/IP 协议栈, 而且要让它较为稳定的工作, 控制丢包率, 必须要进行周密的考虑。目前, 国内有很多人在 8 位机上编写了简化的 TCP/IP 协议栈, 但是内存占用量较大, 运行起来也都或多或少的出一些问题; 而且, 由于规范性不是很高, 让使用者阅读吃力、修改困难, 移植性差。本文用 8051 (当然也可以是其他的 8 位机) 和网络芯片 RTL8019 构建了一个以太网连接设备, 成功地移植 uIP 协议栈到系统上, 并编写了简单的 Telnet 服务程序, 实现了高稳定性的远程监控。目前, 该系统在应用中。

2 系统硬件设计

由于该系统是一个网络读卡器, 有多个模块共同工作, 这里仅介绍网络模块部分。

硬件平台的 CPU 使用 SILICON 的 C8051F022, 该 CPU 的最高处理速度可达 25MIPS, 满足网络应用的需要。处理器与 RTL8019 之间并行连接, C8051F 的 D0-D7 与 RTL8019 的 SD0-SD7 相连, 因为本文所使用的寄存器多分布在 0x00-0x1F 之间, 所以 C8051F 的低 5 位地址线与 RTL8019 的低 5 位地址线连接。RTL8019 的片选与一个 I/O 口连接, 中断 0 与 C8051F 的 P0-P3 的一个 I/O 管脚相连, 然后设置交叉开关使得该管脚为外部中断^[1]。当然, 也可以用查询方式来处理 RTL8019 的事务。

3 RTL8019 驱动设计

RTL8019AS 是一种高度集成的以太网芯片, 能简单地

实现 PNP 并兼容 NE2000 等特性, 这里只用了它极少数的功能。它的驱动包括三个主要的函数: 初始化函数 Eth_Init (); 接收函数 Eth_Read (); 发送函数 Eth_Send ()。RTL8019 的控制是通过寄存器读写来完成的, 与 NE2000 相关的寄存器放在前 3 个页内, 每页的第一个寄存器 CR 都指向同一个地址空间, 其最高两位决定了页号^[2]。

在初始化函数中, 首先要清除以往所有的中断标记, 往 ISR 寄存器 (0 页, 地址 0x07) 写 0xFF; 使 RTL8019 复位, 将复位寄存器 (任何一页, 地址 0x18-0x1F 都映射到复位寄存器) 的值写回复位寄存器, 如果 ISR 寄存器的最高位被置 1, 则说明复位成功。接下来的工作就是设置 RTL8019 的工作模式, 即对各个寄存器写入响应得设置值。RTL8019 内部的 0x4000-0x7FFF 为 RAM 区, 这些区是不能直接访问的, 只能通过 RTL8019 的 DMA 操作来存取。RAM 区用来存放接收和待发送的数据, 读取时是按页来访问的, 页地址为 0x40-0x7F。本文使用页 0x40-0x4B 来存储发送的数据, 用页 0x4C-0x7F 来存储接收到的数据, 因此对寄存器的设置情况为: PSTART=0x4C, PSTOP=0x80, TPSR=0x40。然后设置该网卡芯片的物理地址 (该系统中没有使用 9304 保存物理地址), 根据需要, 设置其它寄存器。

发送的驱动程序编写, 相对来说容易一点, 只管将数据写入发送缓冲区, 然后设置好发送起始地址和发送字节数就可以打开 DMA 进行发送。因为每次在发送之后, 数据区直接被下一次的发送数据覆盖, 因此没有多余的逻辑考虑。但是, 由于接收区是一个循环存储空间, 所以在编写接收驱动程序时, 要注意两点: (1) RTL8019 在接收

时,自动产生了 4 个字节的报头,即在收到一帧数据之后,8019 将当前芯片状态(1 字节)、下一帧数据到来时应该写入的页号(1 字节)、本次接收的总数据量(2 字节)放在了 CURR(1 页,地址 0x07)所指向的那个页的最开头,如果一个页没有写满就完成了一帧的数据接收,下次接收将从下一个页开始存储;(2)恰当地使用 BNRV(0 页,地址 0x03)寄存器是正确访问数据的关键,尤其在数据区快要写满,将要转至开头时,进行适当的逻辑判断是很重要的,当然,最好的办法是参考 RTL8019 所生成那 4 个字节,来对 BNRV 进行设置。

本系统所设计的初始化函数带有一个指针参数,指向存放物理地址的数据区;发送函数带有两个参数,一个为指向待发送数据区头字节的指针,另一个为待发送数据的长度;接收函数只有一个参数:指向接收数据区的指针,但是接收函数有返回值:本次接收到数据帧的总字节数。

4 精简 TCP/IP 协议栈 uIP

uIP 是由瑞士计算机科学院的 Adam Dunkels 等开发的一种免费的、公开源代码的小型 TCP/IP 协议栈,它专为 8 位和 16 位处理器编写。uIP 完全是用 C 语言编写,可移植性强,它在保证一个完整的 TCP/IP 栈的前提下,只保留了最必要的一系列功能,符合 Internet 标准,目前最新版为 uIP0.9。

要搞清楚一个 TCP/IP 协议栈,无非看以下几个方面:数据的输入与输出,即驱动接口;数据在层间的传输;数据包的逻辑判断,如数据包类型的判断、TCP 状态机的转换等;应用层的接口。

uIP 的数据接收与发送都是针对一个全局缓冲区 uip_buf 来进行的,uip_len 记录缓冲区数据的长度。从驱动层到 ip 层(或者 arp 层)再到 TCP 层,都是用不同的结构体指针来操作这个缓冲区的(如图 1),也就是说 uIP 尽

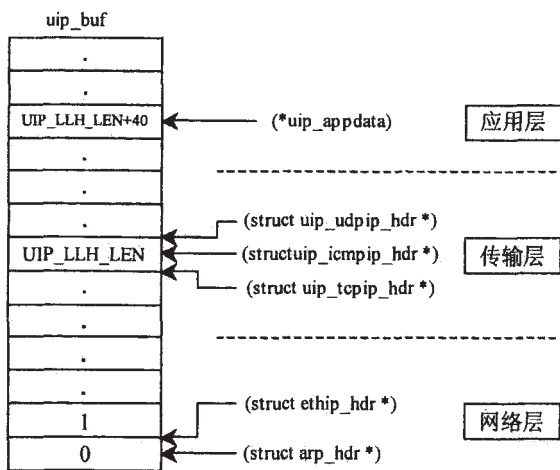


图 1 uIP 的层间数据传递

量保证了 ram 的最少占有量,而且在数据的传递过程中,完全没有数据的拷贝(在编写底层驱动时,本系统也遵循了这个原则)。所有这些设计,都是考虑到 8 位处理器的内存小、速度相对来说很低等诸多因素而专门制定的。

对于收到数据包类型的判断,需要应用程序的干预,即判断该包为 ip 包还是 arp 包,然后调用相应的去报头函数来进行操作。如果为 ip 包,则将全局变量 uip_len 减去 ip 报头的字节数;若为 arp 包,则进行 arp 回复或者更新 arp 表。

uIP 最主要的工作是实现了 TCP。为了减少储存器的使用,uIP 里的 TCP 没有实现发送和接收数据的调整窗口。由于 TCP 是面向连接的协议,因此,uIP 使用了一个结构数组 uip_conn 用于在 uIP 里保持所有 TCP 连接的状态、当地和远端的 TCP 端口编号、远程主机的 IP 地址、重发时间值、上一段重发的编号和连接的段的最大尺寸等。该数组的大小可根据实际应用情况自行定义,节省了空间;数组元素的类型为结构体,其代码如下:

```
struct uip_conn {
    u8_t  tcpstateflags; /*TCP 连接的状态标志 */
    u16_t  lport, rport; /* 本地端口和远端端口 */
    u16_t  ripaddr [2]; /* 远端 IP 地址 */
    u8_t  rev_nxt [4]; /* 期待接收的下一个序列号 */
    u8_t  snd_nxt [4]; /* 上一个发送的序列号 */
    u8_t  ack_nxt [4]; /* 通过从远端的下一个应答去应
    答序列号 */
    u8_t  timer; /* 重发时间 */
    u8_t  nrtx; /* 计算特殊段的重发数量 */
    u8_t  mss; /* 连接的最大段大小 */
    u8_t  appstate [UIP_APPSTATE_SIZE]; /* 应用程序
    的数据区 */
};
```

TCP 的具体代码实现为函数 uip_process 中 tcp_input 标号之后,包括了 TCP 连接状态的转换、TCP 校验、端口监听和检测、序列号和确认号的计算等,并根据接收到的 TCP 首部来判断连接事件的发生与状态。当某个 TCP 事件发生时,改变 uip_flags 为对应状态值,以通知 uIP 的测试函数,并且调用 UIP_APPCALL() 进行应用层的处理。uIP 的轮询与重发也需要应用程序的干预,通过 period-ic_timer 函数来检测 TCP 连接的超时,将状态传递给 uIP 的测试函数,以供应用层知道并进行重发,或者断开连接。

对应用层来说,是通过宏定义 UIP_APPCALL() 来指向应用层函数的,uIP 中的 uip_proces 函数在 TCP 事件发生时,都会调用应用层函数,对于 uIP 来说,相当于是一种回调,即下层协议栈调用上层应用程序。如:#define UIP_APPCALL Telnet_Call。而应用程序通过一系列宏定义来检测连接状态,即上面所提到的测试函数。根据不同的

连接状态, 实施不同的响应。另外, 由于调用 UIP_APPCALL () 之前, 全局指针 uip_conn 是指向当前连接的, 因此, 通过判断当前端口号可以区别不同的应用服务, 从而达到多服务同时工作, 例如: 若值 uip_conn->lport 等于 80, 应用程序可以决定启动一个 HTTP 服务, 值是 23 则启动 TELNET 服务。在应用程序中, 可以通过 uip_send () 来发送数据, 但是需要注意的是: 在一次 UIP_APPCALL () 调用只能使用一次 uip_send ()。

此外, uIP 还实现了 ICMP、UDP 的主动发送, TCP 连接具有流控等功能, 对 uIP 进行配置也非常方便^[4]。

5 uIP 的移植和简单 Telnet 服务的设计、应用

在前面所完成的驱动程序的基础上, 将 uIP 接到那三个函数上还需要一些简单的工作。根据 uIP 的例程, 驱动函数不需要参数, 为了配合 uIP 对驱动程序的接口, 并且考虑到今后可能更换网络芯片和驱动代码, 因此本文设计了一个接口文件, 即将驱动的三个函数进行了封装, 下面是发送和接收两个接口函数的简单描述:

```
#define BUF ((unsigned char *) &uip_buf [0])
int Uip_Eth_Read (void)
{
    int length;
    Eth_SetPage (0);
    if ( (ETH_ISR & 0x01) != 0x01 )
    {
        return -1;
    }
    ETH_ISR = 0x01; /* 如果有接收中断发生, 清除接收中断标记 */
    length = Eth_Read (BUF);
    length -= 4; /* 减去 8019 的 4 字节报头 */
    return length;
}
int Uip_Eth_Send (void)
{
    U16 cnt;
    if (uip_len < 60) /* 数据长度必须大于 46, 加上 14 个字节的 IP 报头, =60 */
    {
        uip_len = 60;
    }
    Eth_Send (BUF, uip_len);
    cnt = 0;
    while ( (cnt++) < 1000)
    {
```

```
        if (ETH_ISR & 0x02)
        {
            ETH_ISR = 0x02; /* 清除发送中断标记 */
            ETH_CR = ETH_CR_START | ETH_CR_NODMA;
            return 0; /* 正常返回 */
        }
    }
    return -1; // 发送出错
}
```

需要注意的是, 发送时, 根据 TCP/IP 协议的要求, 数据长度不能少于 46, 否则, 对方将收不到本地发出的数据^[4]。

在封装之后, 这三个函数就可以直接供 uIP 使用。并且, 更换底层网卡驱动不会影响上层程序。8051 为大端存储模式, 配置时修改宏定义 BYTE_ORDER 为 BIG_ENDIAN。

Telnet 是 TCP 之上比较基本的一个应用协议, 端口号默认为 23, 用它来监控低端系统无疑是一种简便可行的方式。其简单实现的描述为:

```
void Telnet_Init (void)
{
    uip_listen (htons (23)); /* 监听端口 23 */
    初始化 Telnet 状态;
}
void Telnet_Call (void) /* 端口号为 23 时, 调用 Telnet_Call ( ) */
{
    if ( uip_newdata () )
    {
        Telnet 状态改为接收状态;
        回发接收到的字符, 以便客户端显示;
        Telnet_Shellmain (); /* Shell 入口, 根据以往存在 Telnet 缓冲区里的字符串来解析命令 */
    }
    else if ( uip_connected () )
    {
        Telnet 状态改为连接状态, 并做一些初始化工作;
    }
    if ( uip_rexmit () )
    {
        重发刚才要发送的数据;
    }
    return ;
}
```

在 shell 中, 可以自行设定各种命令, 将系统中的各种全局变量发送回客户端或者对其进行修改, 以达到监控的目的。如修改 ip 地址、设备 ID 号, 监视设备温度、记录数据条数、存储空间容量等等。

仅加入网络部分编译之后, 系统的 code 占有量为 12856 字节, ram 使用量为 910 字节, 在 8 位机上运行完全适用。在系统进行远程测试时, ping 通率达到 97% 左右, 这和处理器的速度有关, 延迟时间基本在 30ms 以内, Telnet 连接没有出现断开现象。目前, 该系统已经投入到应用中。总的说来, 通过 Telnet 来实现设备的远程监控是一个较好的选择, 在 8 位处理器上运行 Telnet 也是一种合理的方案。

6 改进

对于少量的网络设备来说, Telnet 方案可行, 但是对于大量设备和标准的网络管理, snmp 才是更为规范的监控

方式。通常来说, snmp 使用 UDP 协议来承载传输报文, 这就要求设备具有 UDP 服务端的功能, 而 uIP0.9 不支持 UDP 端口监听, 因此, 笔者对其进行一定的修改。

参考文献:

- [1] Silicon Laboratories. C8051F02x datasheet [EB/OL]. <http://www.silabs.com>, 2005-02.
- [2] 广州周立功单片机发展有限公司. RTL8019AS 数据手册 (中文) [EB/OL]. <http://www.zlgmcu.com/home.asp>, 2003-02.
- [3] Adam Dunkels. uIP-A Free Small TCP/IP Stack [EB/OL]. <http://www.sics.se/~adam/uip/>, 2002-01-15.
- [4] Richard Stevens. TCP/IP 详解卷 1: 协议 [M]. 北京: 机械工业出版社, 2000.

第一作者简介: 叶勇杰, 男, 1982 年生, 四川简阳人, 硕士研究生。研究领域: 微机保护及变电站综合自动化。

(编辑 梁 玉)

~~~~~

(上接第 31 页)

```
#include "ucos_ii.h"
```

完成了上述移植工作, uC/OS-II 就可以正常运行在 S3C44B0X 微处理器上了。

## 4 中断测试研究

当  $\mu\text{C}/\text{OS-II}$  移植完成后, 必须验证移植的  $\mu\text{C}/\text{OS-II}$  内核能否正常工作, 这一步很关键。测试可以在没有应用程序的情况下进行, 具体包括测试所用的编译器, 测试 OSTaskStkInit () 和 OSStartHighRdy () 函数, 测试中断是否正常。本文重点测试定时中断, 做法是在应用程序中写了两个打印字符串的任务, 通过定时中断来轮流调度执行。在创建的第一个任务中调用下面的 tick\_init 开始时节拍。

```
void tick_init (unsigned char div)
{
    rRTCCON = (unsigned char) 1;
    rTICNT = (unsigned char) (div | 0x80);
    rRTCCON = (unsigned char) 0;
    tick_int_enable (1);
    void tick_int_enable (unsigned char bEnabled)
    { // clear interrupt pending first
        rI_SPIC = (unsigned int) 0x01 << 20;
        rINTMSK &= (unsigned int) ~ (0x01 << 20);
    }
}
```

tick\_init 函数的作用是设置时钟节拍的间隔。tick\_int\_enable 函数作用是清除中断悬挂寄存器中的定时中断位, 接着通过设置 INTMSK 寄存器使能定时中断。

下载到板上, 当看到你的两个任务交替打印字符的

时候, 就说明移植已经成功。在实际应用中要根据情况在每个中断入口处写出对应的中断服务程序, 并且运行一些复杂的任务来进一步验证和提升其系统的实用性。

## 5 结语

RTOS 的使用大大简化了应用程序的设计难度, 也使得程序的可移植性、可扩展性得到了很大的提高。本文完成了实际应用中使用操作系统  $\mu\text{C}/\text{OS-II}$  的最关键的一步, 实际使用时根据需要编写相应的应用程序, 构成实际应用环境中的嵌入式系统。

参考文献:

- [1] Jean J. Labrosse 著. 邵贝贝, 等译. 嵌入式实时操作系统  $\mu\text{C}/\text{OS-II}$  (第二版) [M]. 北京: 北京航空航天大学出版社, 2003.
- [2] 杜春雷. ARM 体系结构与编程 [M]. 北京: 清华大学出版社, 2003.
- [3] Samsung Corp. S3C44B0X Users Manul [EB/OL]. <http://www.samsung.com>, 2000.
- [4] 嵌入式实时操作系统  $\mu\text{C}/\text{OS-II}$  在 ARM 处理器上的实现 [J]. 计算机与现代化, 2005, (3): 120-123.
- [5] S3C44B0X 中 BootLoader 的实现 [J]. 数据采集与嵌入式系统, 2004, (6): 40-43.
- [6] 王田苗. 嵌入式系统设计与实例开发 [M]. 北京: 清华大学出版社, 2002.

第一作者简介: 马 帅, 男, 1981 年生, 山西朔州人, 硕士研究生。研究领域: 嵌入式系统开发。

(编辑 梁 玉)

**Abstract:** In this paper, purpose and function requirement of a chip microcomputer-based measuring instrument for flow characteristics are introduced, and the detailed design of hardware systems and software systems of the measuring instrument is described. The measuring instrument is successfully put into operation.

**Key words:** measuring instrument; electric fuel injectors; chip microcomputer application

#### 006-12-34 *Experimental Research on Bending of Large and Complex "U Shape" Parts*

**LIU Jie, YU Ai-xiang** (Hebi Vocational Technology Institute, Hebi458030, China)

**Abstract:** For changing law of force arm and moment in research process about plate bend, with the help of mathematical operation experiment research, the mathematical model on "U shape" parts for plate bend and the kinematics figure has been set up. When law down the technology on the complicated large bending parts, we utilize this research result to resolve the return bounce on bend.

**Key words:** made die corner; bending moment; retracting springing; bend radius

#### 006-12-37 *Design of the PLC Control for the Hydraulic Power Platform System*

**HE Xiao-yan, ZHAO Ke** (Maoming Collge, Maoming525000, China)

**Abstract:** The paper introduces the programming of the serial communication between the industrial control computer and PLC, using the Spcomm module of Delphi under the Windows environment, by which we design a simple hydraulic sliding platform to dominate and monitor the easy movement of the platform. It aims at improving the controlling elasticity and the automation of the hydraulic sliding platform, and making the administration and supervision of the system more convenient.

**Key words:** Delphi; PLC; serial communication; hydraulic platform

#### 006-12-39 *Research on the Durability Test Facility with High Humidity and Cyclic Temperature Chamber*

**LI Kang-ye, MAI Fan-yuan** (Guangzhou R&D Center of Optics-Mechanics-Electronics Engineering, Guangzhou510630, China)

**Abstract:** The paper introduces a viable scheme of research on the durability test facility with high humidity and cyclic temperature chamber, and the facility is used for high humidity and cyclic temperature durability test for Sealed Insulating Glass Units of construction or refrigeration.

**Key words:** high humidity and cyclic temperature durability test; heating; cooling

#### 006-12-41 *Implementation of Telnet on 8bit Processor*

**YE Yong-jie<sup>1</sup>, GAO Shi-bin<sup>1</sup>, HUANG Jian-min<sup>2</sup>** (1. Department of Electrical Engineering, Southwest Jiaotong University, Chengdu610031, China; 2. Department of Electrical Engineering, Southwest Jiaotong University, E'mei610031, China)

**Abstract:** The connection between embedded system and Ethernet is a hotpot in recent years. The method on 16bit and 32bit processors is quite mature, because of their relatively abundant resources, all of the methods are based on stacks which need a

great deal of ram. But these are not fit for 8bit processors. This article discusses a TCP/IP stack-uIP application on 8bit processors, and designs a steady Telnet server which is used in remote monitoring.

**Key words:** embedded Internet; uIP; TCP/IP stack; C8051F

#### 006-12-45 *The Effects Analysis on Braking Stability of Vehicle Equipped with Retarder*

**Huang Rong-qing, Wu Lei, Shao Jian-hua** (School of Vehicle Engineering, South China University of Technology, Guangzhou510640, China)

**Abstract:** In this paper, the changes of I curve and  $\beta$  line of a truck equipped with a retarder are studied. Especially, making an analysis about the effects which is caused by the changes of I curve and  $\beta$  line. Finally, the allowable maximal retarder braking force is calculated, and we use the allowable maximal retarder braking force for a criterion to evaluate the matching performance of retarder braking force.

**Key words:** retarder; braking force; stability; vehicle

#### 006-12-48 *Reliability Analysis of Double Frequency Electrical Machinery of a Certain Moving Power Station*

**LI Xiao-quan, ZHANG Bai-hua** (The Missile Institute, Air Force Engineering University, Sanyuan713800, China)

**Abstract:** This paper introduced the principle of the double frequency electrical machinery of a certain new weapon system power supply vehicle, and carried on the analysis computation to reliability of double frequency electrical machinery, estimated the "Mean time between failure" of the double frequency electrical machinery, further analysis indicated that the actual reliable level of the double frequency electrical machinery would be higher than the predicted value.

**Key words:** double frequency electrical machinery; reliability; mean time between failures

#### 006-12-51 *Design of the Intelligent Card Water Meter Based on AT90S8515 Microcontroller*

**WANG Jian-xin, LIU Zhi-wen, XUE Miao** (School of Mechanical Engineering, Inner Mongolia University of Science and Technology, Baotou014010, China)

**Abstract:** Flux measuring principle of the intelligent card water meter based on AT90S8515 SCM is presented, the component, the control mode, the hardware and software design and the application are introduced.

**Key words:** Single Chip Microcomputer; intelligent IC water meter; flux control

#### 006-12-54 *USB Interface Design of Hardware and Software on F206DSP*

**ZHANG Lei<sup>1</sup>, XIAO Xing-zhu<sup>2</sup>** (1. Nanjing College of Information Technology, Nanjing210046, China; 2. Electric College of Jiangsu University, Zhenjiang212013, China)

**Abstract:** USB (Universal Serial Bus) is a data transmission, fast, secure, handy and facility. USB interface is introduced in the paper. Emphasis of the paper is laid on design of hardware interface circuit and software.

**Key words:** USB; F206DSP; Universal Serial Bus; serial communication