

The Design and Implementation of Simple TCP/IP Stack Based on ARM Chip

HOU Qun

*School of Physics and Electronic Information Engineering,
Jiangnan University, Wuhan, China*

Abstract: In recent years, research on connecting embedded system with Internet has become one of the focuses in applications of embedded system. The implementation of TCP / IP stack in embedded system is the basis of the research. This paper design a tiny TCP / IP stack suitable for embedded systems with the most extensive 32 embedded RISC processor , high performance ARM7TDMI system. After the standard TCP / IP protocol has been studied and analyzed, according to the characteristic of embedded system and the simplifying principles, some relevant protocols are picked up, simplified and designed. To make sure the essential functions and the irrelevant functions of each protocol, these protocols are analyzed one by one. At last, the implementation models of each protocol are designed. In addition, the method of the network configuration for embedded system is present in this paper. In this paper, a solution with low-cost for networking on embedded system is obtained. Experiments show that it is promising in many fields, such as remote data collecting, industrial controlling.

Keywords: embedded system; TCP/IP protocol stack; UCOS II operating system

基于 ARM 的精简 TCP/IP 协议栈设计与实现

侯 群

江汉大学物理与信息工程学院, 武汉, 中国, 430056

摘 要: 嵌入式系统的 Internet 网络化的研究与应用是近几年来嵌入式应用领域的一个研究热点。嵌入式系统网络化的前提是在嵌入式系统中实现 TCP / IP 协议栈, 本文针对应用最广泛的 32 位高性能嵌入式 RISC 处理器 ARM7TDMI 系统, 设计了一种适合该类系统使用的轻型 TCP / IP 协议栈。首先对标准的 TCP / IP 协议栈功能进行了分析, 然后结合嵌入式系统特点, 根据简化原则, 对相关协议进行选取与简化设计, 确定每个协议的必要功能和可选功能, 得到相应协议的实现模型。系统的嵌入式 Internet 系统解决方案成本低, 性能充分满足网络通信需求, 不仅适用于远程数据采集, 在家居智能化和工业控制等领域也具有广泛的应用前景。

关键词: 嵌入式系统; TCP/IP 协议栈; UCOS-II 嵌入式操作系统

1 引言

嵌入式 Internet 是近几年来随着嵌入式系统的广泛应用和计算机网络技术的进步而发展起来的一项新兴的概念和技术。要想实现嵌入式系统的 Internet 网络化, 就必须在嵌入式系统中实现 TCP / IP 协议栈。嵌入式 TCP / IP 协议栈的研究和应用, 已经成为嵌入式应用领域和网络互联领域的研究热点。本文主要是通过研究基于 32 位微处理器 ARM7TDMI 处理器的嵌入式 TCP/IP 协议的设计与实现, 完成 UCOS-II 在 ARM3000 嵌入式开发平台的嵌入与移植, 结合我们模

拟“智能家电”的实例来完成嵌入式系统的嵌入与移植。实现嵌入式系统联入 Internet 的 TCP/IP, 将多台主机借助交换机与嵌入式系统相连, 可通过操纵计算机主机来控制 RTL8019 网卡, 实现一系列网络应用功能: 修改 mac 地址, 设置 IP, 添加网关, 远程监控等。

2 嵌入式系统中主要处理的以太网协议

TCP/IP 是一个分层的协议。每一层实现一个明确的功能, 对应一个或者几个传输协议。每层相对于它的下层都作为一个独立的数据包来实现。在嵌入式系统中主处理的以太网协议和其所在的分层为 (1) 应用

层(Application)使用的协议是 BSD 套接字(BSD Sockets)。(2)传输层 (Transport)使用的协议是 TCP、UDP。(3)网络层 (Network)使用的协议是 IP、ARP、ICMP、IGMP。(4)数据链路层(Data Link)使用的协议是 IEEE802.3 Ethernet MAC。(5)物理层。

3 TCP/IP 协议简化设计

数据在 TCP/IP 协议栈中双向流动,即接收数据和发送数据两个方向。

3.1 接收网络数据过程

数据源包括网络上对嵌入式系统的访问者和网络管理服务器。数据点包括 ARP 响应处理程序, Ping 相应处理程序和 UDP/TCP 应用程序。数据顺着网络

接口层,网络层,传输层和应用层方向向上流动,在相应的 7 协议层上对协议进行判别,然后调用相应的功能模块对数据进行处理,最后数据到达数据终点,成最终处理。

3.2 发送网络数据过程

与接收过程相反,数据由 Ping 响应处理程序和 UDP/TCP 应用程序产生,数据包括网络上对嵌入式系统的访问者和网络管理服务器。数据顺着应用层,传输层,网络层和网络层的方向向下流动,在相应的协议层上对数据进行封装,形成新的新的数据包,最后由网络设备驱动程序把封装好的数据发送到网络。简化的数据流图如图 1 所示:

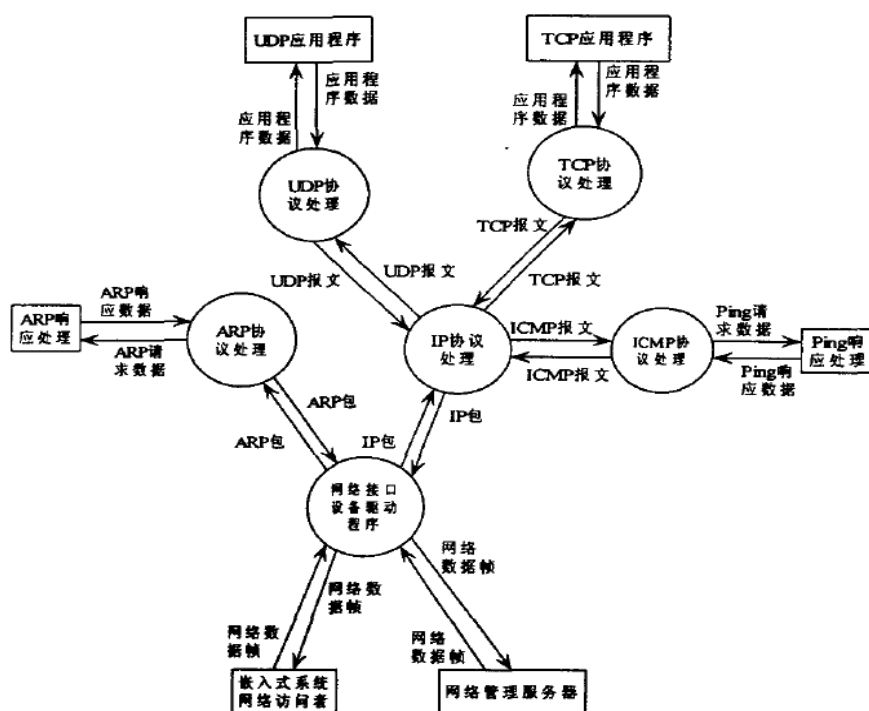


Figure 1. Data flow diagram of simple TCP/IP stack

图 1. 简化的 TCP/IP 协议数据流图

4 网络层协议的实现

对于此 TCP/IP 协议栈简化系统,需要实现的协议有 IP 协议, UDP 协议和 ICMP 协议。

4.1 IP 协议的实现

针对嵌入式应用的特点,本系统的 TCP / IP 协议栈的 IP 块不支持路由器对 IP 模块要求的功能,从而省去了 IP_Forward 程序的实现过程。该 IP 模块也不

支持多 IP 地址,也无需实现 IP 包的分组与重装,这样 IP 模块部分实现起来就比较简单了。它主要由处理 IP 输入的 IP_Input 程序和处理 IP 输出的 IP_Output 两部分组成。

IP-Input 是用来从网络接口芯片接收到 IP 数据包后,首先验证 IP 包的正确性,如果无误则根据 IP 包头部的协议字段进行分发,送到相应的传输层协议进行处理。IP_Input 的处理流程图如图 2 所示:

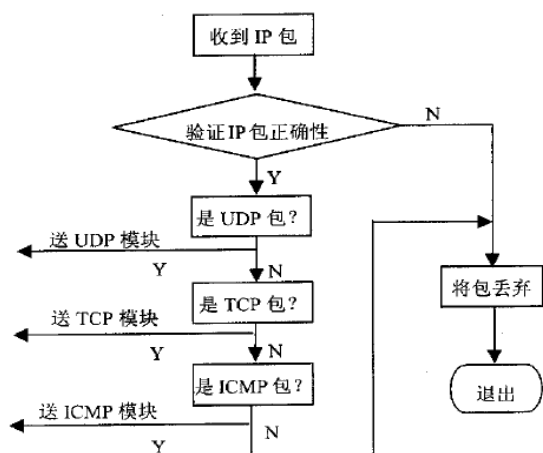


Figure 2. Processing flow diagram of IP_Input

图 2. IP_Input 的处理流程图

IP_Input 中验证 IP 包的正确性是要验证 IP 的版本号是否为 IPv4 和 IP 包的长度是否不小于 IP 包的最小长度, 如果不满足则说明是无效数据包, 应该丢弃。IP 协议字段用于 IP 包分发, 定义如下:

```

#define IP_PROTO_ICMP 1      //ICMP 包
#define IP_PROTO_UDP 17     //UDP 包
#define IP_PROTO_UDPLITE 170 //轻型 UDP 包
#define IP_PROTO_TCP 6      //TCP 包
  
```

IP_Output 的工作是将上层协议送来的数据包加上 IP 头部信息, 选路并发送出去。这里说的 IP 选路是指主机选路而不是路由器的选路, 发送 IP 包前查看发送的目的 IP 地址是否和本机处在同一个子网中, 如果是则直接发送, 否则需要将 IP 包发送到默认的路由器上。

判断 IP 所在的子网是由 Subnet 函数实现的, 它的原理是将目的 IP 地址和子网掩码按位相与, 看相与的结果是否和本地网络号一致, 如果一致则说明目的主机和本机在同一个子网中。

4.2 UDP 协议的实现

UDP 协议模块把上层应用程序传来的数据包加上源端口和目的端口号, 计算校验和并填充, 然后发给 IP 模块即可: UDP 如果接收到 IP 模块送来的数据包只需进行校验, 无误后根据端口号送给上层应用程序即可。这里比较重要的一块就是 UDP 校验和的计算问题。该算法称为网际校验和算法, 它是把被校验的数据每 16 位进行累加, 然后取反, 若数据字节长度为奇数, 则数据尾部补一个字节的 0 以凑成偶数。下面给出适用于本系统的参考程序。

```

unsigned int CalcChecksum
(void * addr, unsigned int count)
  
```

```

register long sum=0;
while (count>1) //以 16bit 为单位求累
加和
{
    sum+=*(unsigned short *)addr++;
    count-=2;
}
//如果整个 UDP 首部为奇数字节, 则加上剩余的
字节
if (count>0)
sum+=*(unsigned char *)addr;
//如果为 32 位, 则压缩为 16 位的和
while (sum>16)
{
    sum = (sum&0xffff) + (sum>>16);
}
return sum; //取反
  
```

需要说明的是该算法是以 UDP 的伪首部 (包括校验和字段) 和数据区为对象进行的, 这和 IP 的校验不同。如果是 UDP 的发送程序, 则先将首部校验和字段置为 0, 求出校验和后赋给校验和字段; 如果是 UDP 的接收程序, 则由于校验和字段是发送端计算后填充好的, 所以如果传输无误的话执行该算法后应得 0, 如果不为 0 则说明传输有误。

4.3 ICMP 协议的实现

ICMP 协议的全称是 Internet 控制报文协议, 它提供了许多功能, 但是嵌入式系统的资源有限。本系统所实现 ICMP 协议, 主要是为了响应对方主机使用的 PING 命令发出的回显请求, 这主要是为对方提供一个测试本地的服务器是否可达的一个手段。本系统的 ICMP 只要求实现很简单的功能, 所以编程很简单。如果收到的是 ICMP 包则 IP 协议模块会将之转给 ICMP 模块处理, ICMP 判断如果是回显请求包的话, 则按要求构造回显应答包, 然后调用底层协议函数发送出去即可。

5 实时操作系统 μC/OS II 下 TCP/IP 协议栈的实现

本系统的 TCP/IP 协议栈是基于瑞士计算机科学院开发的开放源代码的 TCP/IP 协议栈 LwIP, 并结合移植性较好的实时操作系统 μC/OS II 移植到嵌入式开发平台上。LwIP 协议栈在设计时就考虑到了将来的移植问题, 把所有与硬件、OS、编译器相关的部份独立出来, 放在/INC 目录下。因此 LwIP 在 μC/OS II 上的

实现就是修改这个目录下的文件，其它的文件一般不应该修改。

5.1 与 CPU 或编译器相关的 include 文件

/INC 目录下 DEF.H, SYS_ARCH.H, CC.H 中有一些与 CPU 或编译器相关的定义，如数据长度，字的高低位顺序等。这应该与用户实现 μ C/OS II 时定义的数据长度等参数是一致的。

5.2 SYS.H 中与操作系统相关部份

SYS.H 中的内容是与 OS 相关的一些结构和函数，主要可以分为四个部份：

1) sys_sem_t 信号量

LwIP 中需要使用信号量通信，所以在 sys_arch 中应实现信号量结构体和处理函数。由于 μ C/OSII 已经实现了信号量 OS_EVENT 的各种操作，并且功能和 LwIP 上面几个函数的目的功能是完全一样的，所以只要把 μ C/OSII 的函数重新包装成上面的函数，就可以直接使用了。

2) sys_mbox_t 消息

LwIP 使用消息队列来缓冲、传递数据报文，因此要在 sys_arch 中实现消息队列结构 sys_mbox_t，以及相应的操作函数。

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，我们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。

4) sys_thread_new 创建新线程

LwIP 可以是单线程运行，即只有一个 tcpip 线程 (tcpip_thread)，负责处理所有的 tcp/ucp 连接，各种网络程序都通过 tcpip 线程与网络交互。但 LwIP 也可以多线程运行，以提高效率，降低编程复杂度。这时

就需要用户实现创建新线程的函数：

```
void sys_thread_new(void (* thread)(void *arg),
void *arg);
```

5.3 lib_arch 中库函数的实现

LwIP 协议栈中用到了 8 个外部函数，这些函数通常与用户使用的系统或编译器有关，因此留给用户自己实现。如下：

```
u16_t htons(u16_t n); //16 位数据高低字节交换
u16_t ntohs(u16_t n);
u32_t htonl(u32_t n); //32 位数据大小头对调
u32_t ntohl(u32_t n);
int strlen(const char *str); //返回字符串长度
int strncmp(const char *str1, const char *str2, int len); //字符串比较
void bcopy(const void *src, void *dest, int len);
//内存数据块之间的互相拷贝
void bzero(void *data, int n); //内存中指定长度的数据块清零
```

前四个函数通常由用户自己实现。ADT (ARM7) 中，由于使用了 gcc 编译器，gcc 的 lib 库里已经有了后四个函数。如果是其它类型的 CPU，在实现后四个函数时应根据自己的编译器来决定。

5.4 网络设备驱动程序

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。

6 结果

为实现网络远程通信，系统通过 RJ45 接口接入交换机与电脑主机形成小型局域网，其 IP：192.168.0.1，电脑主机为模拟 Internet 的网络终端，IP 为：192.168.0.2。嵌入式开发系统通过 RS232C 的 9 针串口与 PC 机终端相连，对开发系统串口进行相应设置，即可通过串口数据输出对程序流程进行监视。

进行网络通畅性诊断：icmp 协议即用来实现网络状态的诊断功能。用户通过 ping 态命令以及对方 IP 数据的输入，令系统进入 ICMP 数据包的封装和发送过程。过程用串口监视，经过输出显示分析，其状态流程以及数据包封装完全正确，当 PC 电脑向系统发出 ping 命令，若传送到网络上的 IP 数据包未发生

错误, 则 PC 电脑终端应可收到并定会发送回复数据包, 那么系统在接收到回复包之后将会通过显示告知用户对方有回复并显示对方 IP, 依此确定网络的畅通性。

为实现远程操控, 我们设计了一个 VC 程序, 通过菜单“控制—>设置远程 IP”设定 ARM 端 IP。用手指在触摸屏上轻轻地画图, 可以在计算机端显示相应的图形。通过菜单“控制—>远程清屏”命令可以远程清除计算机和触摸屏上的图形, 重新绘图, 实现远程控制。

References (参考文献)

- [1] Xiong Gui Xi,Wang Xiao Hu. Computer Network[M]. Beijing: Tsinghua University Press, 2000: 130–142.
熊贵喜, 王小虎. 计算机网络[M].北京: 清华大学出版社, 2000: 130–142.
- [2] Li Qun Fang,Huang Jian, Technology of Single-Chip Micro-computer and Interface [M](Secondary Edition). Beijing: Electronic Industry Press, 2005: 89–102.
李群芳, 黄建. 单片微型计算机与接口技术[M](第二版). 北京: 电子工业出版社, 2005: 89–102.
- [3] Xie Xi Ren. Computer Network[M] (Third Edition:) DaLian University of Science and Technology Press 2000.
- [4] Li Ming,Kang Jing Qiu,Jia Zhi Ping. Research and Development of TCP/IP Stack [J]. Computer Engineering and Application, 2002, 16: 34–37.
- [5] Wan Jing Hua,Ding Ya Jun. The Connection Of Embedded Equipment and Network with Ethernet Controller [J].Application of Single-Chip Microcomputer and embedded System,2001, 12: 21–24.
万静华, 丁亚军. 以太网控制器的嵌入式设备网络互连[J].单片机与嵌入式系统应用, 2001, 12: 21–24.
- [6] RTL89019AS datasheet <http://www.realtek.com.cn>
uC/OS-II V251 949Crestview Circle Weston, FL 33327 U.S.A.
<http://www.micrium.com>.