

文章编号:1009-1130(2003)01-0001-06

# 基于单片机的嵌入式系统接入互联网技术的研究

邹云峰<sup>1</sup>, 冯 晔<sup>2</sup>

(1. 河海大学 计算机及信息工程学院, 江苏 常州 213022; 2. 河海大学 电气工程学院 江苏 南京 210098)

摘要:首先对当前流行的基于 MCU 嵌入式系统接入 Internet 的解决方案作了一些探讨,并对在 8 位 MCU 上实现 TCP/IP 协议的技术作了分析研究,然后针对 MCU 的特点,提出了一些不同于 UNIX 系统的非常规方法.最后在相应的实验板上基本实现了 TCP/IP 协议,取得了较好的效果.

关键词:单片机;嵌入式系统;互联网;TCP/IP 协议

中图分类号:TN915 文献标识码:A

当前,单片机(MCU)已经在家庭和工业的各个领域得到了应用.但这些嵌入式系统大多数还处于以 MCU 为核心,与一些监测、伺服、显示设备配合,实现一定功能的单独应用阶段.在一些工业应用中,为了实现多个 MCU 之间的信息交流,利用 CAN,RS-232,RS-485 等总线将 MCU 组网,但这是一种局域性网络,有效半径较小,相关通信协议的功能也不完备.因而对于家电和工业的信息化,必须首先完善这些嵌入式系统的网络功能.当前,Internet 已成为社会重要的基础信息设施,是信息流通的主要渠道.所以,基于 MCU 的嵌入式系统接入 Internet 的技术已成为当前研究的热点.

## 1 基于 MCU 嵌入式系统接入 Internet 的解决方案

### 1.1 实现 TCP/IP 协议的技术分析

Internet 的通信协议——TCP/IP 协议和很多网络协议一样,可参照 OSI 参考模型分层,如图 1 所示,它包含的协议繁多,对计算机存储器、运算速度等的要求比较高.

一个系统要接入 Internet,就必须在其上实现 TCP/IP 协议. TCP/IP 协议首先是在 UNIX 系统上得到实现,后来以该技术为蓝本,在 DOS, Windows 和 LINUX 等操作系统上也实现了 TCP/IP<sup>[1]</sup>. 这些 TCP/IP 实现技术的共同特点是:(a)都是基于 32 位或 64 位的 CPU 之上,处理速度快,内存和中断等资源丰富;(b)都是在操作系统上实现的,操作系统为它们提供了诸如:硬

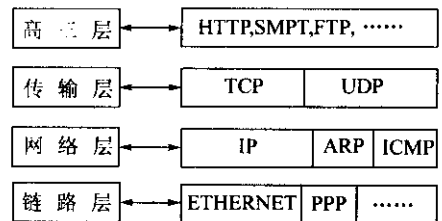


图 1 TCP/IP 协议

Fig. 1 TCP/IP protocol

收稿日期:2002-05-13

基金项目:河海大学常州校区青年教师基金项目资助(河海常青 003).

作者简介:邹云峰(1977—),男,江西靖安人,硕士研究生,通信与信息系统专业.

件驱动、内存管理、中断管理、任务管理等低层的工作,而且操作系统上的良好开发环境对 TCP/IP 的实现也是如有神助。

随后由于专门的网络设备(如路由器)的出现,在 UNIX 上实现 TCP/IP 的原代码也被移植到一些高档嵌入式 32 位 CPU 上,包括内核为 MIPS,ARM 等的 CPU. 这些 CPU 一般都是精简指令集(RISC)内核,处理速度快. 而且在这些嵌入式设备中一般都运行实时操作系统(ROTS),如 WindRiver 公司的 VxWorks,ISI 公司的 PSOS 的嵌入式 Linux 等。

## 1.2 基于 MCU 嵌入式系统接入 Internet 的解决方案

当前的 MCU 大多数都是 8 位的,处理速度和内存先天不足,再加上无操作系统和丰富的硬件接口支持,要较为完整地在一片 MCU 上实现 TCP/IP 协议和系统本身的控制功能几乎是不可能的. 因此,要实现基于 MCU 的嵌入式系统接入 Internet,在技术上有一定难度。

目前采取的策略主要来自两方面的考虑:一是精简 TCP/IP 协议组,尽量省去不必要的协议;二是采用一片已在其上实现了 TCP/IP 的片子,专门充当系统的通信控制器. 当前,比较好的解决方案主要有以下三种。

a. 采用 TCP/IP 专用芯片. 该方案将 MCU 应用系统和内部固化了 TCP/IP 协议的芯片相结合. MCU 应用系统借助 TCP/IP 专用芯片,通过直接拨号或者与以太网相连的方式接入 Internet,硬件电路相对简单,无需其它中间环节的支持. 但由于 TCP/IP 协议实在复杂,没有一个专用芯片会实现所有的协议组,所以这些 TCP/IP 专用芯片都有各自的局限性,并不一定符合用户的具体要求. 另外每一个电子设备都要申请一个 IP 地址,IP 地址资源是有限的,使用时还要付费,成本较高. 采用这种方案的产品主要有 Scenix Semiconductor 公司的 SX-stack,Seiko 公司的 S7600A 和 Rabbit 公司的 Rabbit 2000 芯片等。

b. 利用网关技术. 如图 2 所示,整个系统通过一个网关与 Internet 连接,系统中的 MCU 只使用简单的、与网关通信的协议. 该方案的优点是:(a)由于复杂的 TCP/IP 网络协议是通过网关在 PC 机上实现的,而 MCU 只使用较简单的内部协议,因而对 MCU 的要求相当低,这样就将 MCU 从复杂的协议中解放出来,致力于应用系统的实现;(b)由于有了网关,整个实时控制系统就有了统一的管理,与 Internet 相连只需在网关上设一个 IP 地址,使成本明显降低,同时大大增加了系统的实用性. 目前市场上比较流行 emWare 公司开发的 EMIT (Embedded Micro Internetworking Technology)技术就是采用这一方案。

c. 最原始的也是最直接的方案. 该方案采用一片 MCU 实现 TCP/IP 协议,并作为接入 Internet 的专用通信控制器,它的作用和方案一中的专用芯片一样,将系统接入 Internet. 只不过方案一采用的是已固化的商业化的专用芯片,系统集成方便,工作量小. 而本方案是要自行实现复杂的 TCP/IP 协议,研发周期将大大增长. 但是,由于是自己实现通信协议,就可以针对不同的系统采用不同的协议,这种灵活性非常符合嵌入系统可裁减和量身定做的特点. 另外,对于研究 MCU 系统接入 Internet 的核心技术来说,这是一个不错的方案. 一旦掌握了核心技术,就可根据一些固定的应用,

开发出专用的接入 Internet 的通信产品供用户使用. 基于这些考虑,笔者就针对此方案的实现作了一些研究和探讨。

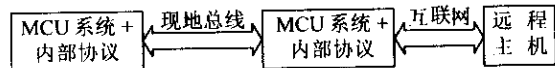


图 2 方案二

Fig. 2 The second solution

## 2 实验板的开发

### 2.1 实验板的总体框架

实验板用于方案三作原理性实验,旨在一片 MCU 上基本实现 TCP/IP 协议,并为接入 Internet 作出测试,如 ping 命令。由于还处于原理性研究,就没有考虑负载用户系统。实验板如图 3 所示,采用了一片 MCU 控制以太网芯片接入以太网,并在这片 MCU 上实现 TCP/IP 协议接入 Internet。

### 2.2 精简 TCP/IP 协议

如上述指出,通常只有在 32 位或 64 位的 CPU 嵌入式系统中才可能较为完整地实现 TCP/IP 协议组。而在 MCU 嵌入式系统的背景下,是难以实现的。不过,一般 MCU 系统的数据量都很小,通信功能也没有必要全之又全,例如,RARP,Proxy ARP,包括 RIP,OSPF 等路由协议,以及高层的 FTP 和 SMTP 等协议就没有必要。因此,实验板撇开了 TCP/IP 协议组的复杂性,采用最精简的必要的协议,以符合嵌入式系统精简实用的特点。

方案三接入 Internet 采用的是以太网接入方式,所以首先必须在链路层上实现以太网的驱动程序。另外在网络层必须实现 ARP,IP 和 ICMP 协议。而在传输层上,主要有 UDP 和 TCP 两个协议,但其设计思想和用途是不一样的。TCP 是面向连接的通信协议,它弥补了 IP 的不可靠性,增加了端到端通信的流量控制和顺序控制的能力,但较为复杂,实现起来较为困难。UDP 是无连接协议,简单,易于实现。对于本实验板,由于是在 8 位的 MCU 上进行,实现 TCP 的难度很大。另外考虑到仪器仪表的网络应用,完全可以直接在 IP 上做应用层的协议,而没必要经过复杂的 TCP 协议。因此,在传输层上,本方案只考虑了简单的 UDP 协议,而撇去了 TCP 协议。对于应用层,本方案还没有涉及到,只是做了一些接收和发送 UDP 包的测试程序。因此本方案实现的协议主要有:ARP,IP,ICMP 和 UDP。

### 2.3 实验板硬件的实现

方案采用的以太网接口芯片为台湾 RealTek 公司制造的 10 M 以太网卡主芯片 RTL8019AS。此芯片与 NE2000 兼容,允许主机选择 16 位或 8 位的总线操作。采用的 MCU 为 51 系列的 89c52 单片机,这是由于该片的片内 ROM 有 4 K,有可能实现复杂的协议。实验板还扩展了一片 64 K 的外部数据存储。实验板原理框图见图 4。

### 2.4 原理板软件的实现

考虑到 TCP/IP 协议和各协议帧结构的复杂性,软件采用 C51 编程<sup>[3]</sup>,这也将增加代码的可移植性。整个程序较长,故仅对整个程序的框架和 TCP/IP 协议的实现方法(特别是不同于 UNIX 上常规的实现方法)作简要的介绍。

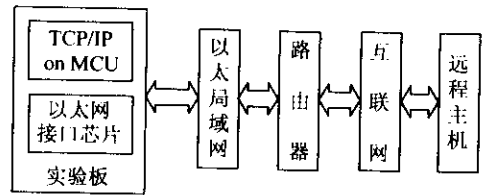


图 3 实验板总体框架和实验环境

Fig. 3 System frame and test environment

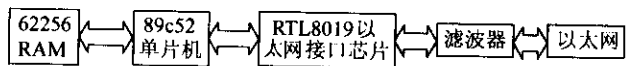


图 4 实验板原理框图

Fig. 4 Block diagram of experiment card

### 2.4.1 程序框架

整个程序主要分:以太网接口芯片的驱动程序、ARP与IP网络层协议的实现、ICMP与UDP传输层协议的实现三个部分。

a. 以太网接口芯片 RTL81019AS 的驱动程序由三个函数组成:芯片初始化函数 `void chipinit()`;以太包接收函数 `void recframe()`;以太包发送函数 `void seframe()`。

b. ARP 协议是处理将 IP 地址转换为物理地址的协议。这部分程序的主要作用是:建立和维护一张地址映射表、ARP 请求和 ARP 应答。程序由两个函数组成:ARP 请求函数 `void arprequest()`;ARP 应答函数 `void arpanswer()`。流程简略框图见图 5。

IP 协议在本方案中是承上启下的协议,所以常常被包括其中,目前尚无很多专门实现此协议的程序,这个协议的实现只是穿插在上层如 ICMP,UDP 等协议中。当然,本程序对一些如 IP 校验、IP 数据包的控制也设计了一些简单的函数。

c. ICMP(网际控制报文协议)的作用是报告处理一个 IP 数据过程中的错误,并提供管理和状态信息。ping 命令就是这个协议提供的一种服务。本程序以实现 ping 命令的回应为代表,说明 ICMP 协议实现的过程。ICMP 处理函数为 `void icmp_handler()`,流程略图如图 6 所示。

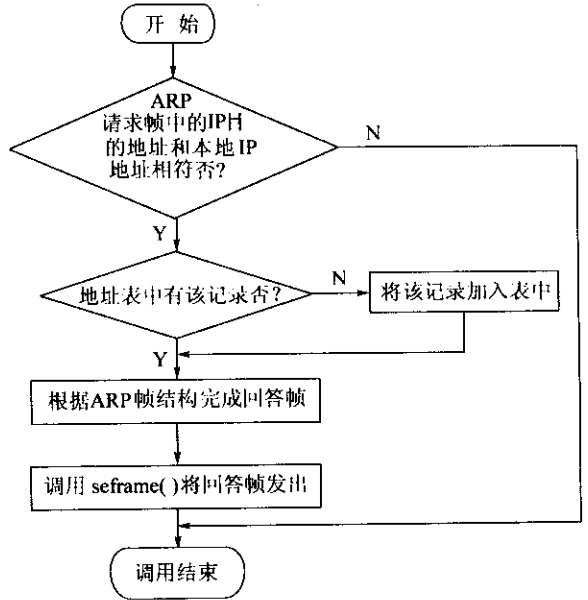


图 5 ARP 应答函数流程图

Fig. 5 Flow chart of ARP answer function

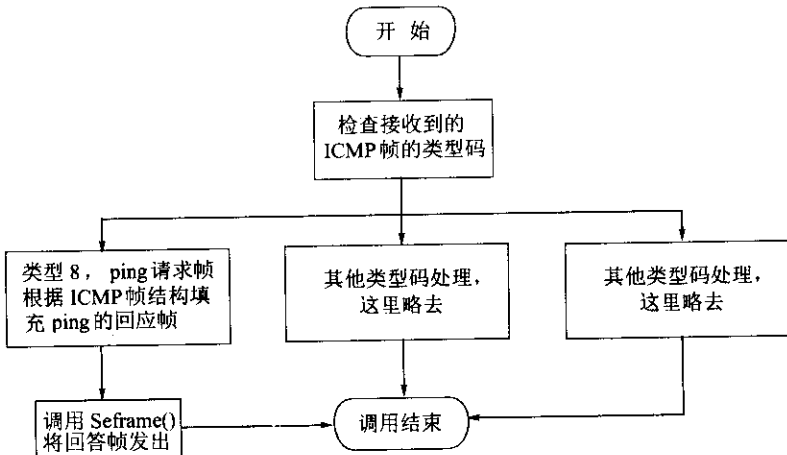


图 6 ICMP 处理函数流程图

Fig. 6 Flow chart of ICMP processing function

## 2.4.2 程序中采用的主要方法

### a. 数据结构的描述.

数据结构设计往往是实现算法的关键,设计得简要合理将非常有助于算法的实现. Windows 或 UNIX 的内存分配是动态的,可根据需要随时分配,随时撤消. 在 UNIX 系统中, TCP/IP 的实现采用的基本数据结构是 mbuf 结构<sup>[2]</sup>. 这是一个用指针实现的链表结构,它指向各个协议包,可动态地增加和减小,极大地方便了 TCP/IP 的实现. UNIX 系统一般运行在 32 位或 64 位的 CPU 上,可访问的内存很大,而且 UNIX 系统提供了方便的动态内存分配接口函数,因此在 UNIX 系统中非常容易实现 mbuf 结构. 而在 51 单片机上却很难用这种方法. 因为 51 单片机内存小,扩展最大也只有 64 K,不可能被各个协议所用,而且 C51 的指针复杂,占用代码太多<sup>[3]</sup>,几乎不可能完成 mbuf 这样复杂的结构的操作.

在本程序中并没有采用 mbuf 复杂数据结构. 这是整个程序的关键之处. 程序中用一个联合体来实现一个所有协议的帧结构体. 这个数据结构并没有按照协议层与层之间的关系定义,而是对程序所要实现的协议帧结构平等地对待. 譬如以太网数据帧应该包括 IP 数帧结构,但这里却是并列的关系. 以下是各协议帧结构联合体 netframe,以太帧结构 ethernet 和 IP 帧结构 ip 的具体表示.

```
union netframe { struct { unsigned char bytebuf[1536]; } bytes; /* 各协议帧结构联合体
                                                         */
                struct ethernet etherframe; /* 以太包 */
                struct arp ariframe; /* ARP 包 */
                struct ip ipframe; /* IP 包 */
                struct icmp icmpframe; /* ICMP 包 */
                struct udp tcpframe; /* UDP 包 */
                };
struct ethernet { unsigned char destadd[6]; /* ethernet 帧中的目的网卡地址 */
                 unsigned char sourceadd[6]; /* 源网卡地址 */
                 unsigned int protocol; /* 下一层协议 */
                 unsigned char packet[1527]; /* 包的内容 */
                };
struct ip { unsigned int head[7]; /* 以太网包头 */
           unsigned char ip_ver; /* 版本, version */
           ..... /* 包头的其他部分,这里省略 */
           unsigned char packet[1517]; /* IP 包的内容 */
           };
```

从以上的表述可以看到,各层帧的结构不是象 UNIX 上实现的那样包含与被包含的关系<sup>[2]</sup>,而是并列的关系. 这样设计的好处在于减少了数据包的存储转移,或者说省去了各层打包和拆包的过程,而且可以不用指针. 譬如,若定义 union netframe newframe,在接收一个数据包时,在链路层(以太包)可用 newframe. etherframe. ... 处理,其后到网络层(如是 IP 包)处理时,就可用 newframe. ipframe. ..., 然后到了传输层,若是 UDP,则可用 newframe.

udpframe. ... 在整个接收这一帧的过程中,各协议层的处理都是在联合体数据 newframe 上进行,各层协议处理完后,程序将结束 newframe 的生命,内存得到释放.所以整个过程都是在同一块数据域中进行的,无需数据包的存储转移,打包和拆包一次性进行(下面将详细论述),同时还省去了用指针对各层协议数据进行变索引,非常方便,非常适合用 C51 来实现.这是本方案的一个最大特点.

#### b. 一次性处理法.

通常实现通信协议的方法是各层处理各层的事务,例如:在发送数据时,高层对本层的数据进行打包后就交给下一层处理;而在接收数据时,低层接收到一帧数据后,根据对本层的帧头分析再交给上一层处理.这种方法层次清晰,易于对复杂通信系统的实现和功能的扩充、删减或更改,符合通信协议的设计理念,也符合软件工程的设计理论.在 UNIX 系统中就是用这种方法实现 TCP/IP 协议的<sup>[2]</sup>,但是这要求处理器处理速度快,数据包要能存储转移.8 位的 MCU 处理速度慢,上述已说明不能实现包的存储转移,所以不应采用这种方法,而应采用一次性处理法.具体说明如下:(a)单任务处理,尽量对一个数据包从网络的低层向高层或从网络的高层向低层一次性处理完,中间除中断外不进行其他的工作;(b)内存上只初始化一个接收帧和一个发送帧,而程序运行中并不形成其他的数据帧.上面介绍的联合结构体为实现这种特殊方法提供了条件.这一处理方法简化了内存的操作和内存的占用,节约了协议间转换处理的时间.

### 3 实验结论

笔者将实验板在图 3 所示的网络环境下作了测试,实现了远程主机对 TCP/IP on MCU 原理板 ping 命令的连接,其中测试的时延一般小于 20 ms,应该说在 MCU 基本实现了 TCP/IP 协议组.

根据实验结果,有如下结论:

- a. 在 MCU 上能够实现 Internet 的以太网接入,或拨号接入(SLIP,PPP 协议都行).虽然整个 TCP/IP 协议组不能全部实现,但能完全满足仪器仪表和家电等信息化所要求的功能.
- b. 由于 MCU 处理能力先天不足,数据传输的速率不高,因此有人提出采用多片 MCU 进行处理,譬如一片接收数据,另一片发送数据,但这样整个板子的设计太烦,程序也不易实现,而且不一定稳定,如此复杂的接入,对于用户值不值也是个问题.
- c. 通过实验已基本掌握 TCP/IP 的单帧单工的技术.要完全实现全双工,只有选用性能更佳的单片机才有可能,譬如 16 位的单片机.但是这将增加系统的成本.另外在单任务的处理环境下,要实现全双工,也是一个值得探讨的问题.

### 参考文献:

- [1] Uyless B. TCP/IP 及相关协议[M]. 良友翻译组译. 北京:机械工业出版社,1998:67-128.
- [2] Gary R, Wright W, Richard S. TCP/IP 详解[M]. 陆雪莹,蒋慧译. 北京:机械工业出版社,2000:2,11-47,162-266.
- [3] 马忠梅. 单片机的 C 语言应用程序设计[M]. 北京:北京航空航天大学出版社. 1991:1,12-256.

## The Integration and Information Exchange of Internal MIS in Colleges

JIANG Bing<sup>1</sup>, SHEN Ye-zhong<sup>2</sup>

( 1. College of International Industry & Commerce, Hohai Univ. ,Changzhou 213022, China; 2. Information Center, Hohai Univ. ,Changzhou 213022, China )

**Abstract:** The paper simply introduces the present condition of the internal MIS in colleges, and points out the heterogeneous databases is the historical problem of the objective esse. How to realize the transparent access is the key for information exchange and system integration. It also sums up the common methods of mutual access of heterogeneous databases and their principle, and brings up a solution based on XML.

**Key words:** manage information system(MIS); heterogeneous database; XML; system integration; information exchange; opening database connectivity(ODBC); middle ware

.....  
(上接第 6 页)

## Study on Accessing Techniques for Embedded System Based on MCU to Internet

ZOU Yun-feng<sup>1</sup>, FENG Ye<sup>2</sup>

(1. College of Computer & Information Engineering, Hohai Univ. ,Changzhou 213022, China; 2. College of Electrical Engineering, Hohai Univ. ,Nanjing 210098, China)

**Abstract:** Firstly, the current solutions of accessing embedded system based on MCU to Internet are depicted, the techniques through which TCP/IP can be loaded into MCU are analyzed and discussed. Then according to MCU features, the unconventional methods being different to UNIX system are presented. Tests with the experiment unit show that TCP/IP has successfully been loaded into MCU and results are satisfactory.

**Key words:** MCU; embedded system; Internet; TCP/IP protocol