东北大学

硕士学位论文

基于工业以太网通信的煤粉喷吹状态监测系统的研究

姓名：邢诚

申请学位级别：硕士

专业：检测技术与自动化装置

指导教师：王玉涛

20060201

基于工业以太网通信的煤粉喷吹状态监测系统的研究

摘 要

以太网技术具有开放性、结构简单、带宽易于扩展、兼容性能好和成本低廉等优点，近几年来在工控领域有了许多成功应用的例子。将工业以太网技术用于煤粉喷吹状态监测系统是一个非常有价值的研究课题。本文针对工业以太网通信系统的硬件与软件等关键技术做了比较全面深入的研究，并成功地将工业以太网通信技术应用在高炉煤粉喷吹状态监控系统中。主要完成了以下工作：

(1) 本文对工业以太网传输技术做了深入的研究，阐述了工业以太网通信系统的几个关键问题，并对它的应用现状和发展前景做了较为全面的总结。

(2) 开发了以 CYGNAL C8051 F021 微控制器和 RTL8019AS 网卡控制芯片为核心的单片机+网卡接口芯片的以太网测控网关。并根据网卡芯片资料开发了相关的网卡芯片驱动软件。

(3) 深入的研究了工业以太网传输协议，并根据本文的传输要求编写嵌入式 TCP/IP 网络协议，实现了通过以太网同时传送支管数据的目的；并且编写了 Web Server 功能，实现利用 IE 浏览器传递数据的目的。

(4) 编写了煤粉喷吹系统状态监测软件，该软件可以显示系统上传的状态数据，如两支管的温度、浓度、速度，还可以显示系统参数设置、温度浓度变化曲线等状态信息，而且根据专家系统中的故障诊断原理，实现了对煤粉喷吹状态监测系统的故障诊断。

实验结果表明，利用以太网通信的煤粉喷吹系统实现了实时的传输数据，即同时获得各个支管的状态信息，并且还能够实现利用 IE 浏览器传递数据，完成远程监视、控制、诊断的功能。研究成果对今后的工业以太网现场仪表的设计与开发具有一定的参考价值。

关键词：工业以太网；监测系统；煤粉喷吹；TCP/IP 协议；故障诊断

Research on Pulverized Coal Injection Monitoring System of Blast Furnace Based on Industrial Ethernet Communication

Abstract

Ethernet technology has the advantages of opening, simple structure, low cost, bandwidth extended easily, good compatibility and cost cheaply. In recent years, there are lots of successful applied examples in industrial control realm. It's a valuable task to apply industrial Ethernet technology in coal powder injecting monitoring system. This dissertation was aimed at the thorough research of the key techniques in industrial Ethernet communication system, such as hardware and software etc, and industrial Ethernet communitcation technique is applied in the pulverized coal injecting monitoring system of blast furnace system successfully. The main contribution and results are as follows:

(1) The industrial Ethernet technology is analyzed in detail. Several key communication problem of industrial Ethernet is expatiated; the unique advantage and industrial application prospect of industrial Ethernet are discussed.

(2) The industry Ethernet control gateway is developed, which the core is the CYGNAL C8051 F021 Microprocessor and the RTL8019AS network card control chip. Moreover, the network card drive software is developed based on the information of network card chip.

(3) The transport protocols of industrial Ethernet are researched deeply, and the embedded TCP/ IP protocols are compiled according to the transport requirement. Achieving the purpose of different pipes can send the data of at one time, and the communication speed is improved. This dissertation achieved the Web Server function, so the state data can be transferred using IE browser.

(4) The software of pulverized coal injection monitoring system of blast furnace is developed. The software can display pipe's status data including temperature, concentration, velocity, further more the fault information can be diagnosed by the software.

The data can be transported real time using Ethernet communication, so pulverized coal injection monitoring system of blast furnace can gain state information of every pipe at the same time. Moreover, the data can be delivered by make use of the browser of IE, which completing the long distance monitoring. The research result has some value in designing the industrial Ethernet spot Equipment for the future.

Key words: industrial Ethernet; monitoring system; PCI; TCP/IP protocol; fault diagnosis

独创声明

本人声明所呈交的学位论文是在导师的指导下完成的。论文中取得的研究成果除加以标注和致谢的地方外，不包含其他人已经发表或撰写过的研究成果，也不包括本人为获得其他学位而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示诚挚的谢意。

学位论文作者签名：邢诚

签 字 日 期：2006.2.25

学位论文版权使用授权书

本学位论文作者和指导教师完全了解东北大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人同意东北大学可以将学位论文的全部或部分内容编入有关数据库进行检索、交流。

（如作者和导师同意网上交流，请在下方签名；否则视为不同意）

学位论文作者签名：　　导 师 签 名：

签 字 日 期：　　签 字 日 期：

第一章 绪 论

1.1 工业以太网概述

近年来，随着计算机、通信、网络等信息技术的发展，传统上用于办公室和商业领域的以太网却悄悄地进入了控制领域[4][5]。不但在工业控制领域管理层和控制层等中上层网络通信中得到了广泛应用，并有直接向下延伸应用于工业现场设备间通信的趋势，成为近年来工业控制网络中新的研究热点[7][8]。以太网与传统现场总线的性能比较如表1.1 所示。从表中可以得出与现场总线相比，以太网具有以下优点[16]：

(1) 能够建立以工业控制网络技术为基础的企业信息化系统。实现过程控制层PCS(Process Control System) 、制造执行层MES(Manufacturing Execution System) 和企业资源规划层ERP(Enterprise Resource Planning)的真正融合。

(2) 成本低廉。硬件产品种类齐全、价格低廉，目前以太网网卡的价格只有 Profibus、FF 等现场总线的十分之一，并且随着集成电路技术的发展，其价格还会进一步下降。

(3) 通信速率高。目前广泛应用的以太网的通信速率为 10Mbps、100Mbps、1000Mbps，10Gbps 以太网技术也逐渐成熟，其速率比目前的现场总线快得多。

(4) 软硬件资源丰富。

(5) 可持续发展潜力大。

目前，以太网用于工业现场设备间通信的研究和产品开发处于初步阶段，还没有一个明确统一的工业以太网的定义和标准，一般来讲，工业以太网是指用于工业测控现场的以太网技术和以太网设备。工业以太网在技术上与商用以太网(IEEE802.3 标准)兼容，但在产品设计时，在材质的选用、产品的强度和适用性方面应能满足工业现场的需要。

表 1.1 以太网与传统现场总线的性能比较[2]
Table1.1 The capability compare between Ethernet and tradition locale bus

项目		以太网	传统现场总线
实时性	确定性	稳定	是
	响应时间	小于等于 4ms	小于等于 5ms
	报文大小	所有规格	有限
可移植性	向上兼容性	高	低
	开放性	高	低
	互操作性	高	低
	标准	IEEE802.3	EN50170/Fieldbus Found
	网络安全性	高（但面临更多问题）	高
	上层协议	TCP/IP	专有协议
控制层次	管理信息层	适用	不适用
	控制层	适用	适用
	设备层	适用	适用
	二进制传感器	网关连接	适用
物理连接	传输介质	光纤、同轴电缆、双绞线、无线	铜线、光纤
	连接设备	控制器、现场设备、远程 I/O	控制器、现场设备、远程 I/O
	最多节点数	64000	较少
	接点间距离	最大 40Km	最大 20Km
	远距离中继器	支持	支持
网络管理	即插即用	支持	不支持
	拓扑结构	总线、星型	总线、星型、环形
	HTTP/WWW	支持	不支持
	SNMP	支持	不支持
性能	速率	10/100/1000Mbps	0-12 Mbps
	升级性能	高	中

1.1.1 标准化情况

对应于 ISO/OSI 七层通信模型，以太网技术规范只映射为其中的物理层和数据链路层；而在其之上的网络层和传输层协议，目前以 TCP/IP 协议为主（已成为以太网之

上传输层和网络层“事实上的”标准），而对较高的层次如会话层、表示层、应用层等没有作技术规定，如图 1.1 所示。

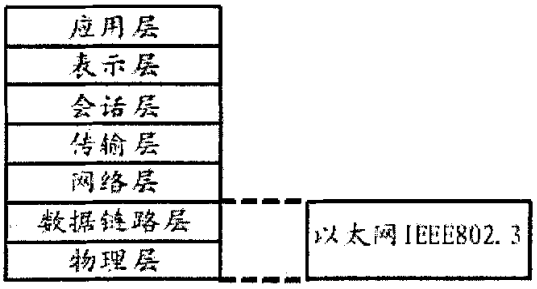

图 1.1 ISO/OSI 七层通信模型和以太网技术规范映射图

Fig.1.1 ISO/OSI seven layers communication model and mapping picture of Ethernet technology criterion

为满足工业现场控制系统的应用要求，工业以太网必须在 Ethernet+TCP/IP 协议之上，建立完整的、有效的通信服务模型，制定有效的实时通信服务机制，协调好工业现场控制系统中实时和非实时信息的传输服务，形成为广大工控生产厂商和用户所接收的应用层、用户层协议，进而形成开放的标准。为此，各现场总线组织纷纷将以太网引入其现场总线体系中的高速部分，利用以太网和 TCP/IP 技术，以及原有的低速现场总线应用层协议，从而构成了所谓的工业以太网协议。典型的有现场总线基金会(Fieldbus Foundation)的 HSE(high speed ethernet),Profibus 国际(Profibus International)的 ProfiNet，ControlNet 国际(ControlNetInternational,CI)和开放 设备网制造商协会(Open DeviceNet VendorAssociation)的 Ethernet/IP,MODBUS 用户集团的 MODBUS/TCP,以及中国的 EPA(Ethernet for Plant Automation)等等[2][23]。

1.1.2 发展应用情况

国外对以太网在控制中的应用较早是用于电力系统的监控[6][9]。美国电力研究院（EPRI）和国际大电网会（CIGRE）的第 35 工作组通过研究都证明：当使用交换式集线器时，10Mbps 以太网完全可以满足实时性要求。自上世纪八十年代开始，国外已经推出不少使用以太网的变电站自动化系统，如美国 GE-Harris 公司的 PowerComm 系统、美国 GE 公司的变电站自动化系统、日本东芝公司的高压变电站分布式测控系统等。DEC 公司开发了一套以太网网络软件产品 DECNET，可以将 DEC 公司生产的各种硬件、软件和通信接口设备连接起来构成各种类型的计算机网络。几个主要的现场总线组织也在开发基于以太网的现场总线协议，一些公司开发了具有以太网接口的仪表、嵌入式以太网控制器及工业以太网交换机等。目前，国际电工委员会 IEC 正着手起草实时以太网(Real-time Ethernet, RTE)标准，旨在推动以太网技术在工业控制领域的全面

应用[18][19]。

八十年代后期，我国电力行业从国外引进的控制系统中就采用了以太网技术，运行证明稳定可靠。这几年，我国的工业以太网技术研究和产品开发呈现蓬勃发展之势。浙江大学、浙大中控、中科院沈阳自动化研究所、清华大学、大连理工大学、重庆邮电学院等单位，在国家“863”计划的支持下，开展了 EPA(Ethernet for Plant Automation)技术的研究，在实时通信、总线供电、远距离传输、网络安全、可靠性技术等领域取得了一批研究成果。其中采用以太网交换技术、全双工通信、流量控制等技术，以及确定性数据通信调度控制策略、简化通信栈软件层次、现场设备层网络微网段化等针对工业过程控制的通信实时性措施，解决了以太网通信的实时性；采用直流电源耦合、电源冗余管理等技术，设计了能实现网络供电或总线供电的以太网集线器，解决了以太网总线的供电问题。起草了 EPA 国家标准，以工业现场设备间通信为目标，以工业控制工程师（包括开发和应用）为使用对象，基于以太网、无线局域网、蓝牙技术+TCP/IP 协议，

起草了“用于工业测量与控制系统的 EPA 系统结构和通信标准”（草案），并通过了由 TC124 组织的技术评审。开发基于以太网的现场总线控制设备及相关软件原型样机，并在化工生产装置上成功应用，该系统自 2003 年 4 月投入使用以来运行一直非常稳定[15]

1.2 工业以太网应用于煤粉喷吹状态监测系统中的意义

目前，虽然与喷煤相关的各项工艺技术（如富氧、高风温、精料）比较成熟，并在高炉冶炼中得到了广泛地应用。但是，高炉喷吹煤粉是一种气力输送过程，在流体力学范围内属于气\固两相流流动，由于流动的非连续性，造成了瞬时流量的波动。如果不能可靠、在线、实时地对进风口前各喷管内高压气流所携带煤粉喷吹状态（如温度、浓度、速度等）进行监测，整个喷吹系统难以维持在合理工况下运行，煤粉在高炉内合理燃烧也就无法实现。这阻碍了高炉煤粉技术获得最大的经济效益[4]。

以往的高炉煤粉喷吹状态监测系统，通讯部分采用的都是现场总线 RS485 通讯协议，对每一支管的状态监测采用的是查询方式，由于支管数量较多，巡检一次的时间较长，不能在同一时刻获得整个系统的状态信息，系统存在时滞性，因此保证不了系统的实时性。

将以太网技术应用到煤粉喷吹状态监测系统中，可以提高原有系统的实时性，即应用以太网技术实现对各支路的检测装置同时进行实时数据的采集。还可以实现利用 IE 浏览器传递数据，完成远程监视、控制、诊断的功能，使企业更快地掌握系统运行状况，对系统合理的正常的工作带来很好的保障。

对系统合理的正常的工作带来很好的保障。

1.3 本文完成的工作和意义

基于以太网(Ethernet)的工业控制网络具有数据传输率高、可靠性好、易维护、可远程传输、互操作性好等优点。随着互联网技术的普及与推广，以太网通信速率的提高、交换技术的发展，使得它受到了全球的拥护和软硬件支持，并得到了迅速发展和普及。因此，基于以太网的工业控制网络是工业控制系统的发展趋势。将工业以太网技术应用于煤粉喷吹状态监测系统是一个非常有价值的研究课题。本文针对工业以太网通信系统的硬件与软件等关键技术做了比较全面深入的研究，并成功地将工业以太网通信技术应用在高炉煤粉喷吹状态监控系统中。主要完成了以下工作：

(1)对工业以太网控制系统做了深入的研究，对它的应用现状和发展前景做了较全面的总结，尤其是着重对工业以太网传输技术进行的研究。

(2)以 CYGNAL C8051F021 微控制器和 RTL8019AS 网卡控制芯片为核心，组成以单片机+网卡接口芯片的以太网测控网关。研究了网卡芯片资料，开发了相关的网卡芯片驱动软件。

(3)研究了工业以太网传输协议，并根据传输要求编写嵌入式 TCP/IP 网络协议，实现了将下位机采集的数据通过以太网传送到负责监控的上位机，上位机再对数据进行相应运算处理。编写了煤粉喷吹状态监测系统软件，完成现场数据的采集与数据处理程序和远程网络通讯程序的开发，并实现 Web Server 功能。实现利用 IE 浏览器传递数据。

(4)研究了专家系统中的故障诊断原理，设计了适用于煤粉喷吹状态监测系统的故障诊断软件。

第二章 工业以太网通信系统的几个关键问题

2.1 实时性问题

在商业应用中，对实时性的要求是软的，对响应时间的要求较低，一般是 2—6s；只要大部分时间满足要求就可以了，偶尔几次不及时响应是关系不大。而工业控制对通信的实时性则要求更高，在理论上至少要满足香农定理，而过程控制对实时性的要求是硬的，因为它常涉及安全，必须在任何时间都及时响应且不允许有任何不确定性。

在工业控制网络中，由于现场设备的地域分散性，现场设备间的信息交互通过网络以信息传递来实现。为了满足监控任务的要求，现场设备间的信息交互必须在一定的通信延迟时间内完成，即必须满足实时性的要求[1]。

随着互联网技术的发展和普及推广，以太网技术在近几年也得到了迅速发展，出现了交换技术、全双工通信、信息优先级、流量控制等一些新技术[11][12]。以太网的通信速率也从 10Mbps 提高到 1Gbps，甚至到目前的 10Gbps，这些都为提高以太网的实时性提供了新的途径。

(1) 限制以太网通信负荷

通过降低通信负荷，可以使以太网上的冲突大大降低，从而有效地提高以太网的通信实时性。实际应用经验表明，对于以太网来说，当通信负荷在 25%以下时，可保证通信畅通；当通信负荷在 5%左右时，网络上碰撞的概率几乎为零。通信负荷主要由节点数目、数据长度和发送频率决定。在过程控制中，数据的发送频率一般是固定的，为 0.5-2s。每个节点发送的数据长度也很小，一般仅为几位或几个、十几个、几十个字节，而且突发性的大量数据传输也很少发生，因此通信负荷的大小主要取决于节点数量。完全可以通过限制每个网段的节点数目，降低网络信息流量。同时，使用 UDP 通信协议，可以充分保证报文传输的有效负荷，避免不必要的数据在网络上传输。

(2) 提高以太网速度

近年来，以太网的通信速率不断提高，目前 10Gbps 以太网的标准也已经推出。高速率意味着对于相同大小的数据，其传输时间大大缩短。同时，以太网通信延迟的大小和冲突发生的次数密切相关，而且与时间槽的长度有关。这使得以太网的通信延迟大大缩短，提高了网络的实时性和确定性。

(3) 采用交换式以太网和全双工通信

在交换式以太网中，交换机将以太网划分为若干个网段，这样就增加了每个网段的吞吐量和带宽。交换机由于具有数据存储、转发的功能，使各端口之间输入和输出的数

据帧能够得到缓冲，不再会发生碰撞；同时交换机还可以对网络上传输的数据的传输只限在本地网段内进行，而不需经过主干网，也不占用其他网段的带宽，从而降低了所有网段和主干网的网络负荷。

全双工通信可以同时发送和接收数据，不仅增加了可利用的宽带(相对半双工，带宽增加了一倍)，而且使数据可以及时传送，减小甚至消除冲突区域，使以太网具有有限的通信延迟，增强了系统的实时性。

(4)采用信息优先级和流量平衡

在工业控制中，将测量和控制信号等实时数据与其他数据进行优先级地划分，首先保证现场实时数据地传送，然后在考虑其他相对次要数据地传送。对于紧急事件信息，可以分配最高地优先权，使其不必进行排队，优先完成传输。主要是通过在 UDP 或 TCP/IP 和 Ethernet 间加一个流量平衡器，作为它们之间的接口，在本地节点，它给实时数据包以优先权来消除实时信息和非实时信息竞争，同时平衡非实时信息，以减少和其它节点实时信息间的冲突。

2.2 安全性问题

用于存在易燃、易爆与有毒气体的工业现场的智能装备以及通信设备，都必须采取一定的防爆措施来保证工业现场的安全生产。现场设备的防爆技术包括两类，即隔爆性(如气密、胶封等)和本质安全型。与隔爆型技术相比，本质安全技术采取抑制点火源能量作为防爆手段，可以带来以上技术和经济上的优点:结构简单、体积小、重量轻、造价低；可在带电情况下进行维护和更换；安全可靠性高；适用范围广。实现本质安全的关键技术是低功耗技术和本质安全防爆技术。

由于目前以太网收发器本身的功耗都比较大，一般都在 60--70mA(5V 工作电压)，因此低功耗的现场设备(如工业现场以太网交换机，传输媒体以及基于以太网的变送器和执行机构等)设计较为难以实现。因此在目前的技术条件下，对以太网系统采用隔爆防爆的措施比较可行。另一方面，对于没有严格的本质安全要求的非危险场合，则可以不考虑复杂的防爆措施。

采用工业以太网技术将会面临其它工控网络所没有的一类网络安全问题。将工业现场控制设备通过以太网连接起来时，由于可以和民用以太网进行对等的连接，并且使用了 TCP/IP 协议，因此可能会受到包括病毒、黑客的非法入侵与非法操作等网络安全威胁，并因此成为众人关心的另一个重要问题。对此，一般可采用网络隔离的办法，将控制区域内部控制网络与外部信息网络系统分开。此外，还可一以通过用户密码、数据加密、防火墙等多种安全机制加强网络的安全管理。

2.3 嵌入式设计问题

以单片机或微控制器(MCU)为核心，与一些监测、伺服、指示设备配合实现一定的功能，通称嵌入式系统。这种将 MCU 嵌入有关的设备中的技术已经在工业的各个领域得到了应用。目前大多数嵌入式系统还处于单独应用的阶段。在一些工业和汽车应用中，为了实现多个嵌入式系统之间的信息交流，利用 CAN，RS-232，RS-485 等总线将嵌入式系统组网，但这种网络的有效半径比较有限，有关的通信协议也比较少，并且一般是孤立于 Internet 以外的。

将嵌入式系统与 Internet 结合起来的主要困难，在于将 TCP/IP 通信协议根据工业应用的特点，加以适当的选择和剪裁并嵌入到工业系统大量应用的以 8 位和 16 位 MCU 为核心的智能节点(测控单元、传感器和执行器等)中，并进而实现嵌入式设备的 Internet 网络化。

嵌入式系统的软件可以采用实时操作系统或简单利用循环加中断结构。部分实时操作系统内部具有对 TCP/IP 协议的支持。有些公司提供了支持 C 语言的 TCP/IP 协议的库函数，这些库函数既可以运行在实时操作系统环境下，也可以直接连接到用户的软件中。

2.4 远程传输和控制问题

由于通用以太网的传输速率比较高(如 10Mbps, 100Mbps, 1000Mbps)，考虑到信号沿总线传播时的衰减与失真等因素，以太网协议(IEEE802.3 协议)对传输系统的要求作了详细的规定，如每一段双绞线(10BASE-T)的长度不得超过 100 米；使用细同轴电缆(10BASE-2)时每段的最大长度为 185 米；而使用粗同轴电缆(10BASE-5)时每段的最大长度也仅为 500 米，对于距离较长的终端设备，可使用中继器(但不超过 4 个)或者光纤通信介质进行连接。

然而，在工业生产现场，由于生产装置一般都比较复杂，各种测量和控制仪表的空间分布比较分散，彼此间的距离较远，有时设备与设备之间的距离长达数公里。对于这种情况，如遵照传输的方法设计以太网络，使用 10BASE-T 双绞线就显得远远不够，而使用 10BASE-2 或 10BASE-5 同轴电缆则不能进行全双工通信，而且布线成本也比较高。同样，如果在现场都采用光纤传输介质，一是现场设备的供电易解决，二是成本同样也非常高，不利于大范围的应用。

为此，在设计应用于工业现场的以太网络时，将控制室与各个拉制域之间用光纤连接成骨干网，这样不仅可以解决骨干网的远距离通信问题，而且由于光纤具有较好的电磁兼容性，因此可以大大提高骨干网的抗干扰能力和可靠性。通过光纤连接，骨干网具

有较大的带宽，为将来网络的扩充、速度的提升留下了很大的空间。各控制域的主交换机到现场设备之间可采用屏蔽双绞线，而各控制域交换机的安装位置可选择在靠近现场设备的地方。

在有些情况下，现场设备离控制域交换机之间的距离确实较长，直接从以太网的集线器端口到现场设备的距离可能超出了以太网规定的距离限制，即信号有可能衰减到不能复原。为了实现在远距离下端对端的可靠连接，必须保证：第一、接收到的信号必须有足够的强度，接收器中的电路能够检测并还原该信号。第二、信号电平必须比噪声电平高，保证接收到的数据才没有错。第三、频率越高衰减越严重。

基于以上考虑，从控制区域到现场设备之间的远离传输采用降低信号传输速率的方法，使现场设备能够正常接收信号。降低速度带来了两个方面的益处：一方面可以减少传输线路对信号的衰减，另一方面还能满足本安型现场设备的要求。根据双绞线的传输特性，传输速率越低，信号衰减越小，传输距离也就越长。事实上，根据 1987 年制订的 1BASE5 Ethernet 标准，数据传输速率为 1Mbps 时每段双绞线(1 类双绞线)的长度可达 250 米。

但降速通信同时也带来了两个不利的方面：一方面是与现场设备通讯的网络性能变差，另一方面由于降速后造成许多网络设备不兼容。然而，由于在工业控制现场，现场仪表需要通信的信息量较小，因此网络吞吐量较小。同时，采用全双工通信和以太网交换机后，由于每个现场设备独享传输介质的带宽，即现场设备的带宽仍然非常高，从网络到设备完全避免了 CSMA/CD 的约束，因此适当范围(如现场设备端通信速率降到 31.25kbps 时，单段网络的通信距离可达 1900m)内的降速传输不会对系统通信响应的实时性产生任何影响。

除此之外，还可以通过进一步改进通信接口，增强其驱动与检测接收能力；采用自适应回波抵消技术，使端口发送出去的较强信号不致对接收到的较弱信号产生干扰，从而使以太网信号的传输距离能够更远。

第三章 工业以太网通讯接口总体设计

3.1 概述

近年来，计算机网络技术发展迅速，以 Internet 和 Intranet 为应用背景的分布式计算机技术也随之受到重视并被日益完善。作为这些技术的一个具体应用，远程监测与故障诊断正开展得如火如荼。特别是随着信息技术和计算机网络技术的发展，远程测控技术正在世界范围内兴起。国内外在基于 Internet 的远程监控系统[10]方面作了大量的研究，对此相关技术的应用有通过 Internet 的原子微探针的远程控制，外科医疗诊断[11]的远程监控系统，表面结构和成分分析的远程监控系统[12]等等。在工业控制领域，人们也正在深入地研究嵌入式技术和网络技术，并努力地将两种技术有机地结合，正是在这样的背景下，本文提出了一种简便的远程监控装置的方案。

本装置为高炉煤粉喷吹状态监测装置，基于 TCP/IP 传输协议，可以在以太网中直接使用。装置主要由以下几部分组成。

(1)现场温度、浓度、速度测量接口：

装置采用 LM35 温度传感器，它的精度可以达到 0.5℃，测量范围为-55℃～150℃，输出电压随环境温度线形变化，可以满足现场要求。

浓度检测是通过螺旋式极板结构的电容传感器，这种结构的传感器具有最均匀的检测场灵敏度分布，其均匀性误差低于 5 %。电容传感器的电容检测线路采用了具有较强抗杂散电容能力的 C/V 转换电路。其灵敏度指标为 1.9 V/pF，非线性度指标为 0.6 %。这完全可以满足现场实际应用的要求。

速度检测也是通过电容传感器完成的。

(2) 以太网通信接口：

电路的设计是采用MCU+网络接口卡(Network Interface Card，NIC)组成网络接口电路。网络接口卡通常称之为“网卡”，网卡的工作原理就是整理计算机上将发送的数据并将数据分解为适当大小的数据包之后向网络发送出去；同时接收通过网络传来的数据给计算机处理。对于网卡而言,每块网卡都有一个唯一的网络节点地址(即48 位物理地址，占6个字节)，这里讨论的网卡是10MB NE2000兼容型以太网卡，它主要有NOVELL公司的NE2000和RealTek公司的RTL8019AS，RTL8139CD等。

用单片机控制以太网网卡进行数据传输，是当前令人感兴趣的一个研究方向，通过单片机控制网卡编程就可以实现局域网内任意终端之间的通信而完全抛开网络操作系统，即在脱离PC环境下实现网卡与其它微处理器之间的接口，从而建立基于非PC 平台

的局域网络。本设计采用Cygnal F021单片机来控制网卡控制主芯片RTL8019AS(以下简称网卡)的，主要研究单片机在以太网卡数据通信中的应用，最终实现煤粉喷吹状态监测装置的网络化，完成对各支路装置监测参数的同时采集。采用这款单片机的主要是因为它内部内置了64K程序存储器，有足够的程序存储空间来存放简化的TCP/IP协议。

RTL8019AS是一种与NOVELL NE2000兼容的8/16位ISA总线网卡芯片，遵循IEEE802.3标准；可实现10BASE-T, 10BASE-2，10BASE-5单芯片解决方案，100脚PQFP封装；支持即插即用、跳线模式；内置16 KB SRAM用于收发缓冲，可降低对主处理器的速度要求；支持AUI、BNC、UTP 多种传输介质的自动侦测。此外,还可以工作在全双工模式下,收发可同时达到10 Mbps；自动产生CRC校验码、数据帧同步码、帧起始定界符，完成以太网数据帧的收发。这些功能可以节省软件的开销，提高系统的执行效率。本设计中设定RTL8019AS 工作在8位模式，即通过一个阻值为27KΩ 的下拉电阻使IOCS16 接地。为了减少连线、降低成本，通过使JP脚接高电平，RTL8019AS 将工作在跳线模式。单片机通过RTL8019AS 收发数据，实际上也就是控制RTL8019AS 的32个端口寄存器完成收发功能，所以只用单片机的几条地址总线与RTL8019AS 的地址总线相连即可。芯片的外部I/O脚TPIN±、TPOUT±通过耦合隔离变压器20F-01接入RJ45，实现与以太网的物理连接。

本装置的结构如图3.1所示。

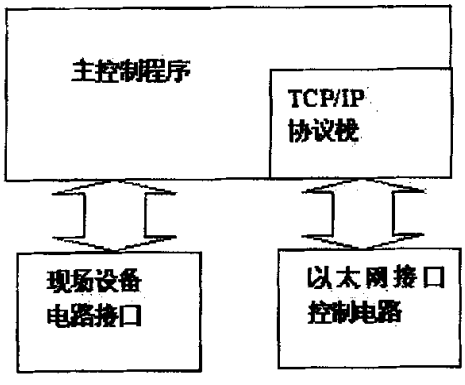

图3.1 高炉煤粉喷吹状态监测装置结构图

Fig.3.1 The structure of blast furnace coal powder blowing state monitoring equipment

(3) 软件部分主要包括数据采集部分和数据通信部分。数据采集部分主要完成对底层传感器信号的采集。主要就是 A/D 采集程序、系统初始化程序等。

A/D 采集主要使用的是单片机内部集成的 A/D 转换单元，初始化程序包括对各个控制参数和片内寄存器的初始化。

装置的通信主要使用 TCP/IP 协议，由于 TCP/IP 协议是一个庞大的协议族，如何在

工业系统中实现 TCP/IP 成为以太网应用于工业系统中的难点。本设计使用的是简化的TCP/IP 协议，主要完成数据的实时传输的任务，在实现基本的功能的同时比较完整地实现了 TCP/IP 协议，该设计不依赖于操作系统，软件完全独立，并具有占用资源少、易于集成、可裁减等优点。

一般使用ISO七层协议中的四层实现TCP/IP协议栈的软件，如图3.2所示。为了有效利用单片机系统有限的资源并提高效率，各层协议的具体实现和通常的TCP/IP协议相比作了较大的简化。

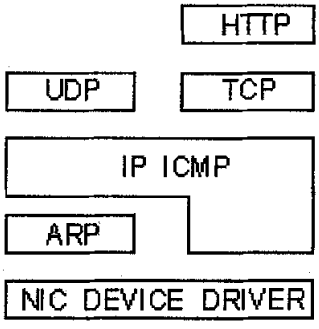

图 3.2 TCP/IP 协议栈

Fig.3.2 TCP/IP protocol

最底层是物理层,它定义了以太网控制器的工作方式,以太网帧的封装和分析、发送以及接收等，本文采用了RealTek公司的以太网控制器芯片RTL8019AS。

第二层是Internet 层，它完成了IP 数据报的封装和IP 头的分析，并根据帧的类型(ICMP、TCP 或UDP 等)进行相应的处理。

第三层是传输层,主要是TCP ，它对Internet 提供点到点服务,在这层上只允许单个连接。它实现了RFC793 中相应的功能,包括三次握手建立连接，各种状态之间的转换,超时重传，连接的撤销等[13]。

最上层是应用层,解决用户特定的应用,而不用关心底层的具体实现。本文实现了HTTP协议，管理者可以利用IE浏览器轻松地获得各传感器的实时数据。

3.2 以太网控制接口

以太网接口[15][18]主要包括两个部分，分别为Cygnal F021单片机电路和RTL8019AS接口电路。下面简单的介绍C8051 F021单片机和RTL8019AS以太网控制芯片。

C8051F 021系列MCU是高精度集成的片上系统。在一个芯片内集成了两个通道的ADC子系统（每个子系统包括一个可编程增益放大器和一个模拟多路选择器）、两个电

压比较器、电压基准、I^2C电路接口、UART、SPI总线接口，5个通用的16位定时器、一个具有5个捕捉/比较模块的可编程计数器/定时器（PCA）、内部震荡器、四个8位通用数字I/O接口和64KB FLASH以及8051兼容的高速微控制器内核。

3.2.1 RTL8019AS 介绍

其引脚图如图3.3所示。RTL8019AS是台湾REALTEK公司生产的以太网控制器，支持IEEE802.3；支持8 位或16位数据总线（由96脚IOCS16B决定）；内置16K的SRAM，用于收发缓冲；全双工，收发同时达到10Mbps；支持10BASET、10BASE2、10BASE5，并能自动检测所连接的介质，在ISA总线网卡中占有相当大比例。RTL8019AS与主机有3种接口模式，即跳线模式、PNP模式和RT模式。本文主要介绍便于嵌入式应用的跳线模式，因此下面主要介绍与跳线模式有关的引脚、寄存器及操作[15]。

(1) 引脚介绍

RTL8019AS可提供100脚的TQFP封装，其引脚可分为电源及时钟引脚、网络介质接口引脚、自举ROM及初始化EEPROM接口引脚、主处理器接口引脚、输出指示及工作方式配置引脚。由于本文主要讨论在非PC环境下的以太网接口该接口不必具有即插即用功能（PNP）和远程自举加载功能，因此不介绍RTL8019AS与自举ROM、初始化EEPROM接口的引脚。其余各部分引脚的功能如表3.1所示。

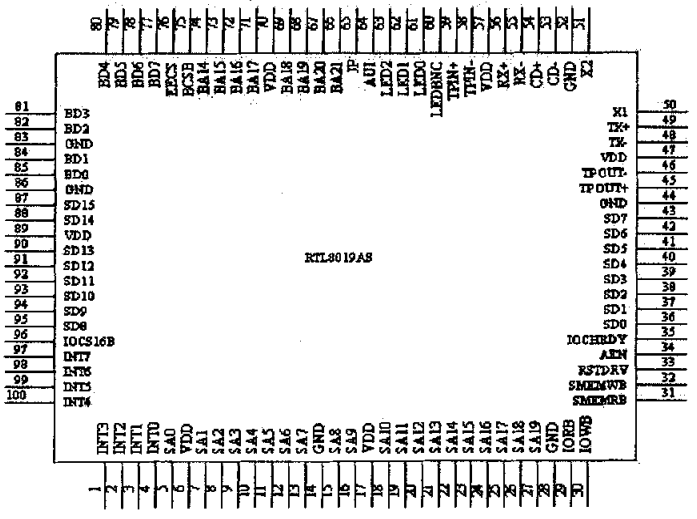

图3.3 RTL8019AS引脚图

Fig.3.3 Pin configuration of RTL8019AS

表3.1 RTL8019AS部分引脚功能表

Table 3.1 General descriptions of the partial pins of RTL8019AS

与网络介质接口引脚		
AUI	输入	用于外部MAU检测
CD+，CD-	输入	AUI冲突，接收来自MAU的冲突
RX+,RX-	输入	AUI接收，接收MAU的输入信号
TX+，TX-	输出	AUI发送，发送MAU的输出信号
TPRX+，TPRX-	输入	从双绞线接收的差分信号
TPTX+，TPTX-	输出	发往双绞线的差分输出信号
与主处理器接口的引脚		
AEN	输入	I/O端口操作允许
INT7—0	输出	中断输出
/IOCS16	输出	16位I/O方式
/IOR，/IOW	输入	端口读写控制
IOCHRDY	输出	I/O通道准备好
/SMEMR，/SMEMW	输入	存储器读写控制
RSRDRV	输入	复位
SA19—0	输入	20位地址总线
SD15—0	I/O	16位数据总线
发光二极管输出引脚		
LEDBNC	输出	介质类型指示
LED0—2	输出	指示控制器的工作状态
工作方式配置引脚		
JP	输入	跳线和非跳线模式选择
PNP	输入	PNP模式选择
IOS3—0	输入	I/O口基地址选择
PL1—0	输入	介质类型选择
IRQS2—0	输入	中断输出

(2)RTL8019AS寄存器介绍

RTL8019AS片内寄存器分为NE2000寄存器组和PNP寄存器组（ 由于PNP方式不适合嵌入式系统使用，这里不再介绍）。NE2000寄存器组地址如表3.2所示。NE2000 寄存器分为4页，都映射到16个I/O端口地址上。主机通过命令寄存器（CR）中的PS0、PS1

位来寻址不同的页，通过16个I/O口地址来寻址页内寄存器。

表 3.2 RTL8019AS寄存器地址表

Table 3.2 Register Tables of RTL8019AS

	PAGE0		PAGE1	PAGE2	PAGE3	
(hex)	[R]	[W]	[R/W]	[R]	[R]	[E]
00	CR	CR	CR	CR	CR	CR
01	CLDA0	PSTART	PAR0	PSTART	9346CR	9346CR
02	CLDA1	PSTOP	PAR1	PSTOP	BPAGE	BPAGE
03	BNRY	BNRY	PAR2	——	CONFIG0	——
04	TSR	TPSR	PAR3	TPSR	CONFIG1	CONFIG1
05	NCR	TBCR0	PAR4	——	CONFIG2	CONFIG2
06	FIFO	TBCR1	PAR5	——	CONFIG3	CONFIG3
07	ISR	ISR	CURR	——	——	——
08	CRDA0	RSAR0	MAR0	——	CSNSAV	——
09	CRDA1	RSAR1	MAR1	——	——	HLTCLK
0A	8019ID0	RBCR0	MAR2	——	——	——
0B	8019ID1	RBCR1	MAR3	——	INTR	——
0C	RSR	RCR	MAR4	RCR	——	——
0D	CNTR0	TCR	MAR5	TCR	——	——
0E	CNTR1	DCR	MAR6	DCR	——	——
0F	CNTR2	IMR	MAR7	IMR	——	——
10-17	远程DMA端口					
18-1F	复位端口					

(3) RTL8019AS与主机的接口模式

RTL8019AS与主机的接口模式有三种， 即跳线模式、PNP模式和RT模式，见表3.3所示。

①跳线模式 这种模式与早期的网络控制器兼容。RTL8019AS的端口基地址、中断口等都由开关或跳线器决定。跳线模式简单，但配置资源麻烦。

②PNP模式 与微软的PNP协议兼容。在这种模式下，RTL8019AS的端口基地址、中断口等都由93C46设定，但需要进行PNP芯片的识别，不便与单片机连接。

③RT模式 为了避免PNP模式下的PNP芯片识别和配置过程，REALTEK公司提供RT模式。在RT模式下RTL8019AS的端口基地址、中断口等也是由EEPROM 93C46设定的。

表 3.3 RTL8019AS的接口模式

Table3.3 The mode of the interface of RTL8019AS

JP引脚	PNP引脚	93c46中的PNP位	配置模式	配置来源	初始化关键字
1	×	×	跳线	跳线器	RT
0	1	×	PNP	93c46	RT和PNP
0	×	1	PNP	93c46	RT和PNP
0	0	0	PNP	93c46	RT

本课题选用的是第一种方式即跳线方式。这样可以省去一个93c46，减少电路连线。同时也节约了硬件开发成本。在原理图中只需要把65脚JP接至+5V即可。第65脚JP是输入引脚，当65脚为低电平时，rtl8019as工作在第2种或第3种方式，具体由93c46里的内容决定。市场购买的rtl8019as网卡一般第65脚为悬空的，rtl8019as悬空时，引脚的输入状态为低电平（其它引脚也是这样，悬空的输入脚的电平为低电平，这是因为芯片里面有一个100k的下拉电阻），网卡工作在第2、3种工作方式时，需要使用93c46芯片。如果把65脚接高电平（VCC），那么网卡的I/O和中断就不是用93c46的内容决定，这时可以不接93c46。那么这时候的I/O和中断IRQ向量是多少，这时需要用到64，65，78，79，80，81，82，84，85等引脚。芯片的I/O地址由引脚85，84，82，81（IOS3 -- IOS0）决定如表3.4 所示。本文选中的基地址是300H，即把上述的四个引脚（IOS3 -- IOS0）接地。芯片的中断线由以下引脚80，79，78（IRQ2--IRQ0）决定，见表3.5 所示。本设计中使用的是查询方式。在UNIX或WINDOWS里，对网卡驱动无一例外都是采用中断方式。而在单片机的应用中，采用的方案是查询式的。因为电脑的处理速度快，一次中断的处理时间也很短，不会影响系统内的其它中断。但在单片机里就不行了，处理一次中断，收取一个数据包一般要消耗几毫秒的时间，这将封锁其它中断的产生（只有高优先级的中断可以执行），而单片机往往还存在其它一些中断，比如串口接收中断，A/D条件中断、键盘中断等需要被执行，这就使得消耗时间长的网卡中断改为查询方式执行。在电脑里，对网卡的驱动相对简单，而在单片机里需要处理的事情更多。比如缓冲区溢出，阅读一些驱动程序源代码，你可能发现在电脑里的一些程序根本没有处理溢出的代码。因为电脑执行快，网卡缓冲区的溢出几乎是不会发生的。电脑往往采用即插即用的方式来驱动网卡，而单片机却不能这样做。因为即插即用的方式实现起来需要很多代码，单片机一般使用跳线方式和8位DMA方式[15]。

表3.4 RTL8019AS基地址选择寄存器

Table 3.4 Select I/O base address

IOS3	IOS2	IOS1	IOS0	I/O BASE
0	0	0	0	300
0	0	0	1	320
0	0	1	0	340
0	0	1	1	360
1	0	0	0	380
1	0	0	1	3A0
1	0	1	0	3C0
1	0	1	1	3E0
0	1	0	0	200
0	1	0	1	220
0	1	1	0	240
0	1	1	1	260
1	1	0	0	280
1	1	0	1	2A0
1	1	1	0	2C0
1	1	1	1	2E0

表3.5 RTL8019AS中断选择寄存器

Table 3.5 IRQ interrupt Select register

IRQ2	IRQ1	IRQ0	中断名	对应ISA中断IRQ
0	0	0	INT0	IRQ2/9
0	0	1	INT1	IRQ3
0	1	0	INT2	IRQ4
0	1	1	INT3	IRQ5
1	0	0	INT4	IRQ10
1	0	1	INT5	IRQ11
1	1	0	INT6	IRQ12
1	1	1	INT7	IRQ15

(3) RTL8019AS与传输介质的连接

RTL8019AS可与双绞线或同轴电缆接口，连接方法如图3.4 所示。介质选择由引脚PL0、PL1决定。另外，RTL8019AS还自动测试介质连接是否成功。图3.4中，NS8392是同轴电缆驱动/接收器，其电源应与RTL8019AS的电源隔离，一般使用一个DC/DC 电源转换器得到该隔离电源。16PT-005A内有3个耦合变压器，用来传输信号，同时抑制来自介质的共模噪声/干扰。20F-01N是双绞线驱动/接收器，内部也有2个传输变压器。64脚

AUI，该引脚决定使用AUI还是嵌入式BNC（包括RJ45）接口。常用的网卡的接口一般很少用AUI。嵌入式BNC接口方式支持8线双绞和同轴电缆。高电平时使用AUI接口，悬空为低电平，使用BNC（双绞线）接口。因此将该引脚悬空即可。

网络接口类型由74，77(PL0,PL1)引脚决定，见表3.6所示。使用第一种自动检测就可以了，这种设置将自动检测使用的是同轴电缆还是双绞线[15]。

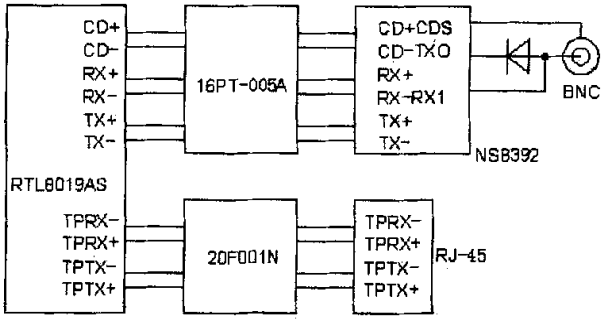

图3.4 RTL8019AS与传输介质的连接图

Fig.3.4 RTL8019AS linked with network medium types

表3.6 网络接口类型选择

Table 3.6 Network medium types

PL1	PL0	介质类型
0	0	TP/CX自动检测 (10BASET link test enable)
0	1	10BASET with link text disable
1	0	10BASE-5
1	1	10BASE-2

3.2.2 C8051F021 接口设计

3.2.2.1 C8051F021 主要特点

(1)模拟外设[18]

①逐次逼近型ADC0

12位ADC，可编程转化速率可达100KSPS；有8路外部输入，可选择为单端或差分输入；可编程的放大器增益：16，8，4，2，1，0.5；同时还包括数据相关窗口中断发生器，内置温度传感器。

②8位ADC1

可编程转化速率可达500KSPS；有8路外部输入，可编程的放大器增益：4，2，1，0.5。

③两个12位的DAC，可以同步输出产生无抖动波形。

④两个模拟比较器，可用于产生中断或者复位。

⑤电压基准，包括内部基准和外部基准输入。

⑥精确的VDD监视器和降压检测器

(2)片内JTAG调试和边界扫描

片内调试电路提供全速、非侵入式的在线系统调试；支持断点、单步、观察点、堆栈监视器、支持观察修改存储器和寄存器。

(3)高速8051微控制器内核

含有流水线指令结构，70%的指令执行时间为一个或者两个系统时钟周期，速度可达25MIPS，同时有22个矢量中断源。

(4)存储器

有4352B内部RAM（4KB+256B），64K FLASH存储器接口，可以在系统编程；外部有64K数据存储器接口（可以使用在地址复用和非复用的方式）。

(5)数字外设

4个8位的I/O，所有的I/O口均耐5V电压；可以同时使用硬件I^2C、SPI以及两个UART串口；有可编程的PCA阵列，5个通用的16位通用定时器/计数器，同时还有专门的"看门狗"定时器。

(6)时钟源

内部可编程震荡器和外部震荡器（晶体、RC、C或者外部时钟）；同时还有RTC方式选择。C8051F021引脚图如图3.5所示。

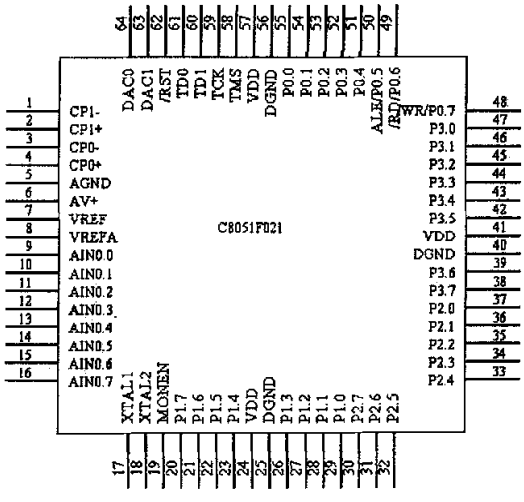

图3.5 C8051F021引脚图

Fig.3.5 Pin configuration of C8051F021

3.2.2.2 C8051F021 单片机介绍

(1) 存储器组织

如图3.6所示。C8051F021存储器包括三部分：程序存储器、数据存储器和外部地址数据空间。其中程序存储器为在线可编程的64K空间，地址为0x0000~0xFFFF；数据存储器除了包括标准的8052的256B的内部数据RAM之外，还包括位于CIP-51外部数据存储器空间4K的RAM块（XRAM）。F021同时还提供64K的外部数据存储器接口。其中标准的8052的256B的内部数据RAM与8052单片机的结构一样，这在使用中是十分方便的。而位于CIP-51外部数据存储器空间4K的RAM块为片内的XRAM，对一般的应用来说不需要使用外部扩展RAM了。

数据存储器除了包括标准的8052的256B的内部数据RAM之外，还包括位于CIP-51外部数据存储器空间4K的RAM块（XRAM）。F021同时还提供64K的外部数据存储器接口。配置外部存储器接口主要有以下5个步骤：

①将 EMIF 选到端口（P3，P2，P1，P0）。

②选择复用或者非复用方式。

③选择存储器模式（只使用片内、不带块选择的分片方式、带块选择的分片方式或只用外部存储器）。

④设置与片外存储器或外设接口的时序。

⑤选择所期望的相关端口的输出方式。

其中前三个步骤是由寄存器EMIOCF来选择的，第四步是由寄存器EMIOCN来设置的，详细的资料请参照参考文献[18]。

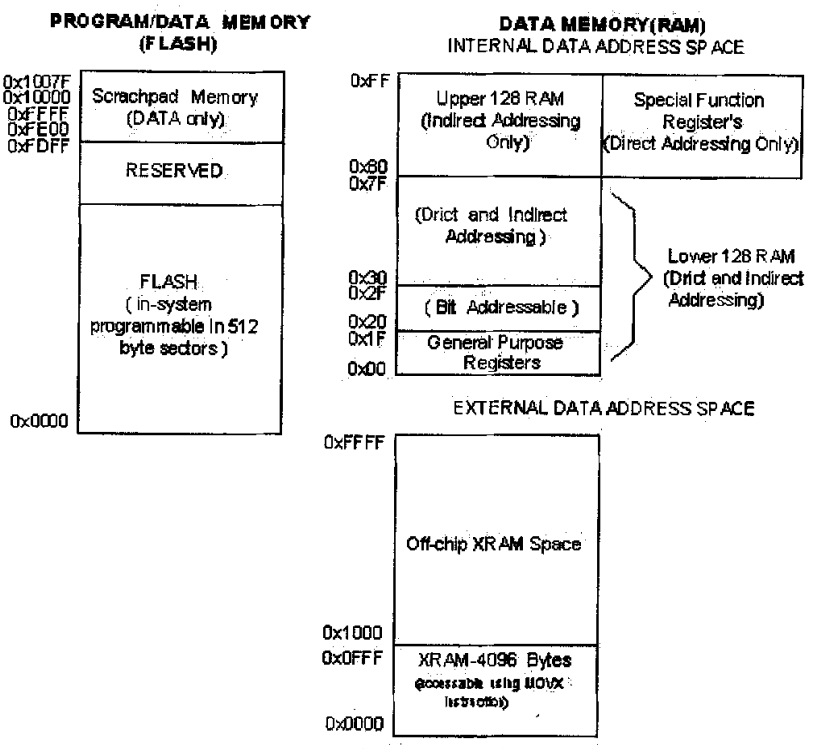

图3.6 C8051F021存储器组织结构

Fig.3.6 Configuration of C8051F021 storages

(2) 端口的输入和输出

C8051F021有8位端口组织的32个数字I/O引脚,端口（P3，P2，P1，P0)。所有的引脚都耐5V电压，都可以配置成漏极开路或者推挽输出和弱上拉。端口的配置主要分为以下几个要点。

①交叉开关引脚配置

②配置端口为输出方式

③配置端口为数字输入方式

④弱上拉

⑤配置端口为模拟输入方式

详细资料请参照参照参考文献[18]。

(3) 模/数转换器

C8051F021包括两套ADC子系统即ADC0和ADC1。本设计中主要使用的是ADC0。参考电压采用的是外部电压基准。使用REF0CN寄存器配置电压基准，选择引脚VREF作为ADC0的电压基准。

(4) 可编程计数器阵列

C8051F021单片机内部有个可编程计数器阵列（PCA）。

3.2.3 以太网接口设计

由于篇幅有限，本文原理图位于附录中，见图A.1。

由图可知：单片机和以太网控制芯片的连接主要是三总线的连接。这里有几点需要说明[19]。

(1) 由于C8051F021单片机为8位数据总线结构，在与RTL8019AS连接时必须将RTL8019AS的96脚IOCS16接地，一般通过一个27K的电阻连接。

(2) RTL8019AS的复位引脚与C8051F021的P0.0连接。

(3) RTL8019AS的20根地址总线主要是为了读写自举ROM，对单片机来说只要使用16根地址总线就够了；RTL8019AS的31、32脚接VCC，这样可以屏蔽了远程自举加载功能；AEN接GND，使能I/O操作。

(4) 单片机工作在数据总线和地址总线非复用的方式，P3口做数据总线，P1口和P2口做地址总线。这样可以省去一片地址锁存器，减小电路板的面积。

(5) 由于选择RTL8019AS的基地址为300H，本设计中采用的是以下的连接方式。

表3.7 地址连接方式

Table 3.7 Address connection types

8019	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10-A19
V/G						GND	GND	GND			GND
F021	A0	A1	A2	A3	A4				A15	A15	

表中V表示VCC，G表示GND。当A15为高电平时将选中RTL8019AS的基地址，即当单片机片外地址为8000H时将对应RTL8019AS的基地址300H。

(6) 本装置主要应用于以太网，应用的接口主要是RJ45，采用10BASE-T布线标准通过UTP非屏蔽双绞线与以太网通讯，而RTL8019AS内置了10BASE-T收发器，所以网络接口的电路比较简单[19]。芯片的外部I/O脚TPIN±（接收线）、TPOUT±（发送线）通过耦合隔离变压器20F-01接入RJ45与RX±、TX± 相连接,实现以太网的物理连接。

3.3 TCP/IP 协议栈的实现

随着Internet 网络软、硬件的快速发展，互联网用户正在以指数增长。以太网经过20多年的发展，已经成为当今互联网络中底层链接不可缺少的部分；TCP/IP协议栈[17]是一个非常复杂和庞大的系统[3]，它是Internet互联网安全可靠通讯的重要组成部分，通常在通用计算机系统和操作系统下实现。而各种家电设备、PDA、仪器仪表、工业生产中数据的采集与控制等设备也正在逐渐地走向网络化，以便共享互联网络中庞大的资

源。在某些应用领域，嵌入式设备在价格、体积及实时性等方面,有着通用计算机无法比拟的优点，因此，嵌入式设备的网络化开发有着广阔的前景[20]。为了实现各种小型设备能够通过互联网进行远程监控，除了在网络硬件接口建设方面的开发以外，一个重要的问题就是根据嵌入式系统的资源有限，功能单一的特点开发一套微型TCP/IP互联网络协议。如何在价格低廉、资源有限的嵌入式系统的环境中实现网络通讯功能，已经成为嵌入式网络开发人员面对的重要问题。本文设计的微型TCP/IP协议栈能满足这一需要，能够完成常用的网络通讯功能。

3.3.1 通用 TCP/IP 协议

3.3.1.1 总体介绍

网络协议通常分不同层次进行开发，每一层分别负责不同的通信功能。一个协议族，比如TCP/IP[21]，是一组不同层次上的多个协议的组合。TCP/IP通常被认为是一个四层协议系统，如图3.7所示。每一层负责不同的功能：

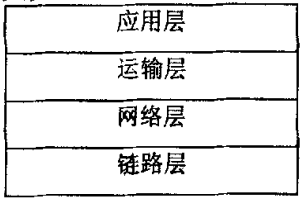

图 3.7 TCP/IP协议族的四个层次

Fig.3.7 Four layers of TCP/IP protocol

(1) 链路层，有时也称作数据链路层或网络接口层，通常包括操作系统中的设备驱动程序和计算机中对应的网络接口卡。它们一起处理与电缆（或其他任何传输媒介）的物理接口细节。

(2) 网络层，有时也称作互联网层，处理分组在网络中的活动，例如分组的选路。在TCP/IP协议族中，网络层协议包括IP协议（网际协议），ICMP协议（Internet互联网控制报文协议），以及IGMP协议（Internet组管理协议）。

(3) 运输层主要为两台主机上的应用程序提供端到端的通信。在TCP/IP协议族中，有两个互不相同的传输协议：TCP（传输控制协议）和UDP（用户数据报协议）。TCP为两台主机提供高可靠性的数据通信。它所做的工作包括把应用程序交给它的数据分成合适的小块交给下面的网络层，确认接收到的分组，设置发送最后确认分组的超时时钟等。由于运输层提供了高可靠性的端到端的通信，因此应用层可以忽略所有这些细节。而另一方面，UDP则为应用层提供一种非常简单的服务。它只是把称作数据报的分组从一台主机发送到另一台主机，但并不保证该数据报能到达另一端。任何必需的可靠性必

须由应用层来提供。

(4) 应用层负责处理特定的应用程序细节[22]。几乎各种不同的TCP/IP实现都会提供下面这些通用的应用程序：

①Telnet 远程登录

②FTP 文件传输协议

③SMTP 简单邮件传送协议

④SNMP 简单网络管理协议

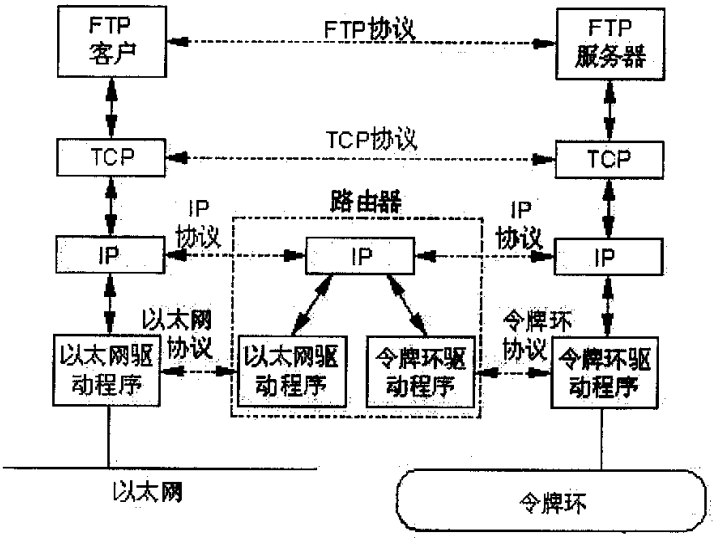

图3.8 通过路由器连接的两个网络

Fig.3.8 Two networks linked by a Router

假设在一个局域网（LAN）如以太网中有两台主机，二者都运行FTP协议，图3.8列出了该过程所涉及到的所有协议。这里列举了一个FTP客户程序和另一个FTP服务器程序。大多数的网络应用程序都被设计成客户—服务器模式。服务器为客户提供某种服务，本例中就是访问服务器所在主机上的文件。在远程登录应用程序Telnet中，为客户提供的服务是登录到服务器主机上。在同一层上，双方都有对应的一个或多个协议进行通信。例如，某个协议允许TCP层进行通信，而另一个协议则允许两个IP层进行通信。在图3.8的右边，可以看到应用程序通常是一个用户进程，而下三层则一般在（操作系统）内核中执行。尽管这不是必需的，但通常都是这样处理的，例如UNIX操作系统。在图3.8中，顶层与下三层之间还有另一个关键的不同之处。应用层关心的是应用程序的细节，而不是数据在网络中的传输活动。下三层对应用程序一无所知，但它们要处理所有的通信细节。

在图3.8中列举了四种不同层次上的协议。FTP是一种应用层协议，TCP是一种运输层协议，IP是一种网络层协议，而以太网协议则应用于链路层上。TCP/IP协议族是一组不同的协议组合在一起构成的协议族。尽管通常称该协议族为TCP /IP，但TCP和IP只是其中的两种协议而已（该协议族的另一个名字是Internet协议族(Internet Protocol Suite)）。网络接口层和应用层的目的是很显然的——前者处理与通信媒介有关的细节（以太网、令牌环网等），而后者处理某个特定的用户应用程序（FTP、Telnet等）。

3.3.1.2 **以太网和 IEEE802.3 封装**

在TCP/IP协议族中，链路层[23]主要有三个目的：

(1) 为IP模块发送和接收IP数据报；

(2) 为ARP模块发送ARP请求和接收ARP应答；

(3) 为RARP发送RARP请求和接收RARP应答。TCP/IP支持多种不同的链路层协议，这取决于网络所使用的硬件，如以太网、令牌环网、FDDI（光纤分布式数据接口）及RS-232串行线路等。由于本设计是在以太网中使用，这里仅仅讨论以太网协议。

以太网这个术语一般是指数字设备公司(Digital Equipment Corp.)，英特尔公司(Intel Corp.）和Xerox公司在1982年联合公布的一个标准。它是当今的TCP/IP采用的主要的局域网技术。它采用一种称作CSMA/CD的网络控制方法，其意思是带冲突检测的载波侦听多路接入（Carrier Sense, Multiple Access with Collision Detection ）。它的速率为10 Mbps，地址为48 bit。

采用IEEE802.3标准，以太网的物理传输帧格式如图3.9所示。

PR	SD	DA	SA	TYPA	DATA	PAD	FCS
56位	8位	48位	48位	16位	不超过1500字节	可选	32位

图3.9 以太网的物理传输帧格式

图.3.9 Ethernet physical frame

PR 同步位，用于接收双方的同步时钟，同时也指明了传输的速率；10M和100M的时钟频率不一样，但是100M网卡可以兼容10M网卡。

SD 分隔位，表示下面跟着的是真正的数据，而不是同步时钟，为8位的10101011，跟同步位不同的是最后两位是11而不是10。

DA 目的地址，以太网的地址是48位二进制地址，表明该帧传输给哪个网卡，如果是FFFFFFFFFFFF，则是广播地址，广播地址的数据可以被任何网卡接收。

SA 源地址，48位。表明该帧的数据是哪个网卡发送的，即发送方的网卡地址。

TYPE 类型字段，表明该帧的数据是什么类型的数据，不同的协议的类型字段是不

一样的；如0800H表示该数据为IP包，0806H表示该数据为ARP包，814CH表示该数据为SNMP包，8137H表示该数据为IPX/SPX包。

DATA 数据段，该段数据不能超过1500个字节，因为以太网规定整个传输包的最大长度不能超过1514字节，其中14字节包括DA、SA、TYPE。

PAD 填充位，由于以太网的帧传输的数据包最小长度不能小于60字节，除去DA、SA、TYPE 14个字节，还必须传输46字节的数据。当数据段的数据长度不满足46个字节时后面补000000...(当然也可以补其它值)。

FCS 32位数据校验位。为32位的CRC校验，该校验有网卡自动计算。自动生成并且自动在数据段的后面填入。

3.3.1.3 ARP 地址解析协议

当一台主机把以太网数据帧发送到位于同一局域网上的另一台主机时，是根据48bit的以太网地址来确定目的接口的。设备驱动程序从不检查IP数据报中的目的I P地址。地址解析为这两种不同的地址形式提供映射：32bit的IP地址和数据链路层使用的任何类型的地址。ARP为IP地址到对应的硬件地址之间提供动态映射。

在以太网上解析IP地址时，ARP请求和应答分组的格式如图3.10所示（ARP可以用于其他类型的网络，可以解析IP地址以外的地址。紧跟着帧类型字段的前四个字段指定了最后四个字段的类型和长度）。以太网报头中的前两个字段是以太网的源地址和目的地址。目的地址为全1的特殊地址是广播地址。电缆上的所有以太网接口都要接收广播的数据帧。两个字节长的以太网帧类型表示后面数据的类型。对于ARP请求或应答来说，该字段的值为0x0806H。形容词hardware (硬件)和protocol (协议)用来描述ARP分组中的各个字段。例如，一个ARP请求分组询问协议地址（这里是IP地址）对应的硬件地址（这里是以太网地址）。硬件类型字段表示硬件地址的类型。它的值为1即表示以太网地址。协议类型字段表示要映射的协议地址类型。它的值为0x0800即表示IP地址。它的值与包含IP数据报的以太网数据帧中的类型字段的值相同，这是有意设计的。接下来的两个单字节的字段，硬件地址长度和协议地址长度分别指出硬件地址和协议地址的长度，以字节为单位。对于以太网上IP地址的ARP请求或应答来说，它们的值分别为6和4。操作字段指出四种操作类型，它们是ARP请求（值为1）、ARP应答（值为2）、RARP请求（值为3）和RARP应答（值为4）。这个字段必需的，因为ARP请求和ARP应答的帧类型字段值是相同的。接下来的四个字段是发送端的硬件地址（在本例中是以太网地址）、发送端的协议地址（IP地址）、目的端的硬件地址和目的端的协议地址。注意，这里有一些重复信息：在以太网的数据帧报头中和ARP请求数据帧中都有发送端的硬件地址。对

于一个ARP请求来说，除目的端硬件地址外的所有其他的字段都有填充值。当系统收到一份目的端为本机的ARP请求报文后，它就把硬件地址填进去，然后用两个目的端地址分别替换两个发送端地址，并把操作字段置为2，最后把它发送回去。

协议地址长度

以太网目的地址	源地址	帧类型	硬件类型	协议类型			OP	发送端地址	发送端IP	目的地址	目的IP
6	6	2	2	2	1	1	2	6	4	6	4

硬件地址长度　　（单位 字节）

图3.10 用于以太网的ARP请求或应答格式

Fig.3.10 Formats of Ethernet ARP request or answer

在TCP / IP协议通讯中，涉及到的地址是IP地址（IPv4为32bit），这是来自网络层的地址，然而以太网都有自己的寻址机制（为48bit地址），所以两层之间必须进行地址之间的转换。向以太网中发送IP数据时，首先根据目的IP地址在ARP高速缓存表中查询相应的以太网地址。如果查到匹配的结点，则相应的以太网地址被写入以太网帧首部，数据报被加入到输出队列等候发送。如果查询失败，ARP会先保留待发送的IP数据报，然后广播一个询问目的主机硬件地址的ARP报文，等收到回答后再将IP数据报发送出去。RARP协议恰恰相反，它负责将以太网地址转化为IP地址。

3.3.1.4 IP 网际协议

网络层主要涉及到 IP 协议、ICMP 协议和 IGMP 协议。IP 协议是 TCP/IP 中的重点，所有的应用都要通过它在 Internet 进行数据传输，IP 协议提供的是不可靠、无连接的数据分组传送服务。不可靠性是指 IP 协议中不保证 IP 数据分组能成功的到达目的地，但是它提供很好的传输服务，它有一个简单的算法：当发生错误时，它就丢弃数据分组，并发送 ICMP 消息给信息源端。无连接是指 IP 协议中不维护任何关于后续数据分组的信息，数据分组之间是独立的，从源端口到目的端口的路径也是不相关的。ICMP 协议主要是用于差错控制，所涉及的错误信息包括目的地不可达到、报文超时、无效头字段、源端拥塞、路由重定向、回声请求和回声应答、时间标记请求和时间标记应答、地址掩码请求和地址掩码应答等等。IGMP 协议主要是用于支持主机和路由器并行多播。

IP是TCP/IP协议族中最为核心的协议。所有的TCP、UDP、ICMP及IGMP数据都以IP数据报格式传输。不可靠（unreliable）的意思是它不能保证IP数据报能成功地到达目的地。IP仅提供最好的传输服务。如果发生某种错误时，如某个路由器暂时用完了缓冲区，IP有一个简单的错误处理算法：丢弃该数据报，然后发送ICMP消息报给信源端。任何要求的可靠性必须由上层来提供（如TCP）。无连接（connectionless）这个术语的

意思是IP并不维护任何关于后续数据报的状态信息。每个数据报的处理是相互独立的。这也说明，IP数据报可以不按发送顺序接收。如果一信源向相同的信宿发送两个连续的数据报（先是A，然后是B），每个数据报都是独立地进行路由选择，（可能选择不同的路线），因此B可能在A到达之前先到达。

IP数据报的格式如图3.11所示。普通的I P首部长为20个字节，除非含有选项字段。最高位在左边，记为0 bit；最低位在右边，记为31 bit。4个字节的32 bit值以下面的次序传输：首先是0～7 bit，其次8～15 bit，然后16～23 bit，最后是24~31 bit。这种传输次序称作big endian字节序。由于TCP/IP首部中所有的二进制整数在网络中传输时都要求以这种次序，因此它又称作网络字节序。以其他形式存储二进制整数的机器，如little endian格式，则必须在传输数据之前把首部转换成网络字节序。目前的协议版本号是4，因此I P有时也称作IPv4。首部长度指的是首部占32 bit字的数目，包括任何选项。由于它是一个4比特字段，因此首部最长为6 0个字节。服务类型（TOS）字段包括一个3 bit的优先权子字段（现在已被忽略），4 bit的TOS子字段和1 bit未用位，但必须置0。4 bit的TOS分别代表：最小时延、最大吞吐量、最高可靠性和最小费用。4 bit中只能置其中1 bit。如果所有4 bit均为0，那么就意味着是一般服务。

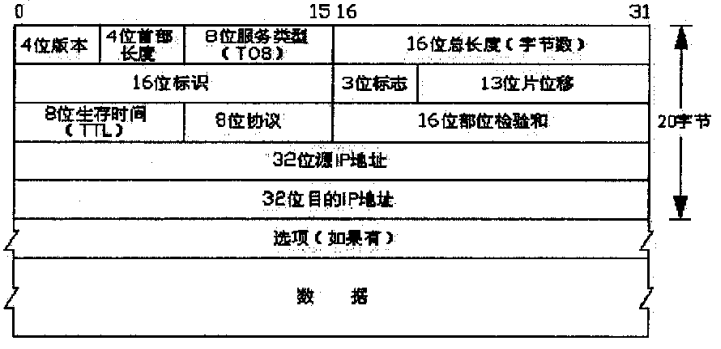

图3.11 IP数据报格式及首部中的各字段

Fig.3.11 Formats of IP Protocol Datagram and Datagram Header

总长度字段是指整个IP数据报的长度，以字节为单位。利用首部长度字段和总长度字段，就可以知道IP数据报中数据内容的起始位置和长度。由于该字段长16比特，所以IP数据报最长可达65535字节。标识字段唯一的标识主机发送的每一份数据报。通常每发送一份报文它的值就会加1。TTL（time_to_live）生存时间字段设置了数据报可以经过的最多路由器数。它指定了数据报的生存时间。TTL的初始值由源主机设置（通常为32或64），一旦经过一个处理它的路由器，它的值就减去1。当该字段的值为0时，数据报就被丢弃，并发送ICMP报文通知源主机。首部检验和字段是根据I P首部计算的检验

和码。它不对首部后面的数据进行计算。ICMP、IGMP、UDP和TCP在它们各自的首部中均含有同时覆盖首部和数据的检验和码。

3.3.1.5 UDP 用户数据报协议

UDP是一个简单的面向数据报的运输层协议：进程的每个输出操作都正好产生一个UDP数据报，并组装成一份待发送的IP数据报。这与面向流字符的协议不同，如TCP，应用程序产生的全体数据与真正发送的单个IP数据报可能没有什么联系。UDP数据报封装成一份IP数据报的格式如图3.12所示。

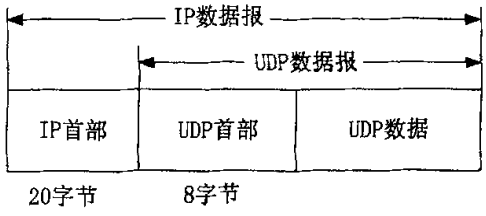

图3.12 UDP封装

Fig.3.12 UDP-assembled datagram

UDP首部的各字段如图3.13所示。

端口号表示发送进程和接收进程。由于IP层已经把IP数据报分配给TCP或UDP（根据IP首部中协议字段值），因此TCP端口号由TCP来查看，而UDP端口号由UDP来查看。TCP端口号与UDP端口号是相互独立的。尽管相互独立，如果TCP和UDP同时提供某种知名服务，两个协议通常选择相同的端口号。这纯粹是为了使用方便，而不是协议本身的要求。UDP长度字段指的是UDP首部和UDP数据的字节长度。该字段的最小值为8字节（发送一份0字节的UDP数据报是OK）。这个UDP长度是有冗余的。UDP检验和覆盖UDP首部和UDP数据。UDP和TCP在首部中都有覆盖它们首部和数据的检验和。UDP的检验和是可选的，而TCP的检验和是必需的。

16位源端口号	16位目的端口号
16位UDP长度	16位UDP校验和
数据（如果有）	

图3.13 UDP首部

Fig.3.13 UDP Protocol Datagram Header

3.3.1.6 TCP 传输控制协议

尽管TCP和UDP都使用相同的网络层（IP），TCP却向应用层提供与UDP完全不同

的服务。TCP提供一种面向连接的、可靠的字节流服务。面向连接意味着两个使用TCP的应用（通常是一个客户和一个服务器）在彼此交换数据之前必须先建立一个TCP连接。

TCP数据被封装在一个IP数据报中，如图3.14所示。

图3.15显示TCP首部的数据格式。如果不计任选字段，它通常是20个字节。

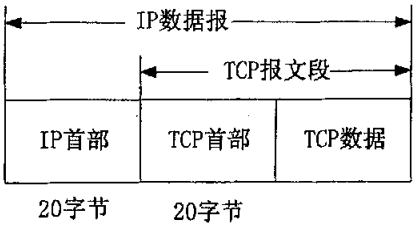

图3.14 TCP协议在IP协议中的封装

Fig.3.14 TCP-assembled datagram

源端口			目的端口
序号			
确认序号			
首部长度	保留	标志	窗口
校验和			紧急指针
可选项			
数据			

图3.15 TCP包首部

Fig.3.15 TCP Protocol Datagram Header

每个TCP段都包含源端和目的端的端口号，用于寻找发送端和接收端应用进程。这两个值加上IP首部中的源端IP地址和目的端IP地址唯一确定一个TCP连接。有时，一个IP地址和一个端口号也称为一个插口（socket）。插口对（socket pair）(包含客户IP地址、客户端口号、服务器IP地址和服务器端口号的四元组)可唯一确定互联网络中每个TCP连接的双方。序号用来标识从TCP发端向TCP收端发送的数据字节流，它表示在这个报文段中的的第一个数据字节。如果将字节流看作在两个应用程序间的单向流动，则TCP用序号对每个字节进行计数。序号是32 bit的无符号数，序号到达$2^{32}-1$后又从0开始。当建立一个新的连接时，SYN标志变1。序号字段包含由这个主机选择的该连接的初始序号ISN（Initial Sequence Number）。该主机要发送数据的第一个字节序号为这个ISN加1，因为SYN标志消耗了一个序号。既然每个传输的字节都被计数，确认序号包含发送确认

的一端所期望收到的下一个序号。因此，确认序号应当是上次已成功收到数据字节序号加1。只有ACK标志为1时确认序号字段才有效。

发送ACK无需任何代价，因为32 bit的确认序号字段和ACK标志一样，总是TCP首部的一部分。因此，看到一旦一个连接建立起来，这个字段总是被设置，ACK标志也总是被设置为1。

TCP为应用层提供全双工服务。这意味数据能在两个方向上独立地进行传输。因此，连接的每一端必须保持每个方向上的传输数据序号。TCP可以表述为一个没有选择确认或否认的滑动窗口协议。我们说TCP缺少选择确认是因为TCP首部中的确认序号表示发方已成功收到字节，但还不包含确认序号所指的字节。当前还无法对数据流中选定的部分进行确认。例如，如果1～1024字节已经成功收到，下一报文段中包含序号从2049～3072的字节，接收端并不能确认这个新的报文段。它所能做的就是发回一个确认序号为1025的ACK。它也无法对一个报文段进行否认。例如，如果收到包含1025～2048字节的报文段，但它的检验和是错误的，TCP接收端所能做的就是发回一个确认序号为1025的ACK。TCP的流量控制由连接的每一端通过声明的窗口大小来提供。窗口大小为字节数，起始于确认序号字段指明的值，这个值是接收端正期望接收的字节。窗口大小是一个16 bit字段，因而窗口大小最大为65535字节。检验和覆盖了整个的T C P报文段：TCP首部和TCP数据。这是一个强制性的字段，一定是由发端计算和存储，并由接收端进行验证。最常见的可选字段是最长报文大小，又称为MSS (Maximum Segment Size)。每个连接方通常都在通信的第一个报文段（为建立连接而设置SYN标志的那个段）中指明这个选项。它指明本端所能接收的最大长度的报文段。

TCP的状态变迁图如图3.16所示。在这个图中要注意的第一点是一个状态变迁的子集是“典型的”。粗的实线箭头表示正常的客户端状态变迁，粗的虚线箭头表示正常的服务器状态变迁。第二点是两个导致进入ESTABLISHED状态的变迁对应打开一个连接，而两个导致从ESTABLISHED状态离开的变迁对应关闭一个连接。ESTABLISHED状态是连接双方能够进行双向数据传递的状态[21][22]。

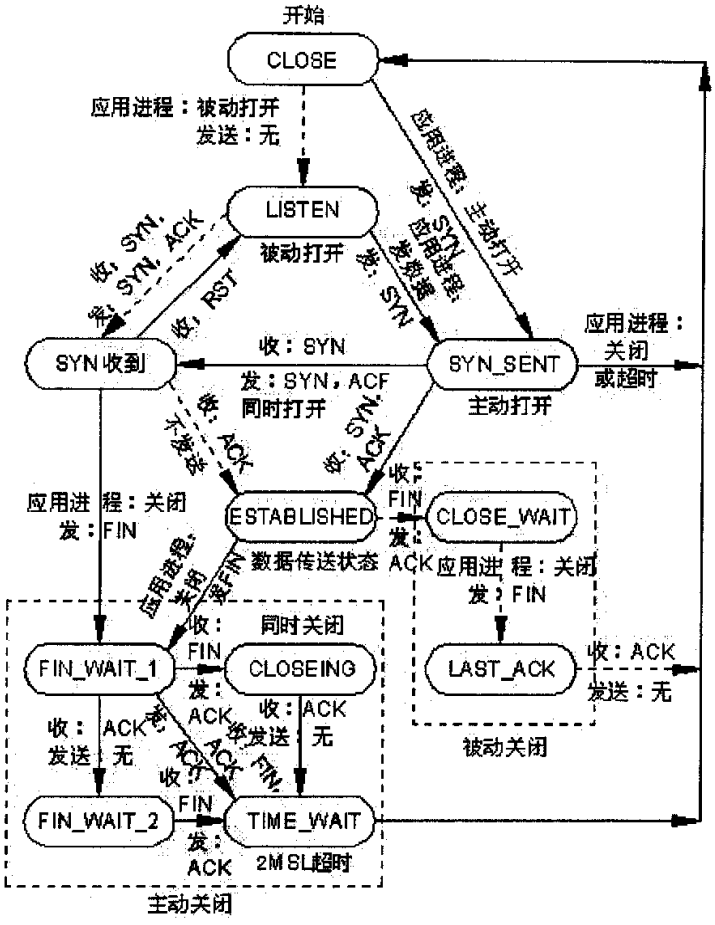

图3.16 TCP的状态变迁图

Fig.3.16 TCP Protocol state variance

3.3.2 微型 TCP/IP 协议栈的实现

在这里将介绍嵌入式系统中的TCP/IP的实现[24][25]。这里主要介绍的几个协议包括：IP协议、UDP协议、ARP协议、TCP协议。

（1）ARP 协议的实现

在通信系统中，ARP协议主要负责将节点IP地址转换到对应的实际物理地址。在下位机向上位机发送数据时，首先要将目的IP地址转换为实际的目的物理地址。因为在互连网中，是使用实际的物理地址来区分各个节点的。因此，只有将IP地址与物理地址一一对应起来才能完成数据的传输。

在上一节中已经论述了通用的ARP协议的实现与基本功能。但是在微型TCP/IP协议中，由于使用场合主要是以太网，仅仅需要ARP地址解析就能够完成任务了。考虑两台计算机A和B共享同一个物理网络的情况。每台计算机都有一个IP地址和一个唯一的以太

网地址，假如A要通过物理网络向B发送一个IP包，A仅仅知道B的IP地址，那么如何实现通信呢？这时就需要ARP协议来解析了。从IP地址向物理地址的变换是通过查表来实现的。如果IP模块在ARP表中找不到某一目标的IP地址的对应项，那么它就通过一个广播发送ARP请求来寻找目标。收到广播的每个ARP模块检查请求分组的目标IP，当自己的IP和该地址相同时，就直接发送一个响应分组给源以太网地址的主机。如果源主机没有收到ARP响应，那么ARP表中也就没有其登录项，本地IP将抛弃发向这个目标的IP分组。这就是ARP工作的流程图。当实现IP地址向以太网地址的转化时一般采用动态映射的方法，即在使用过ARP协议的主机上保留一个高速缓存表，最好短时间内就能更新。当一个主机收到ARP应答时，它就保留这个IP地址映射。

本设计主要使用以下几个模块来实现ARP协议。

①ARP发送 ②ARP接收 ③ARP地址解析 ④ARP更新缓冲表

各个模块的流程图如下。

①ARP发送和接收

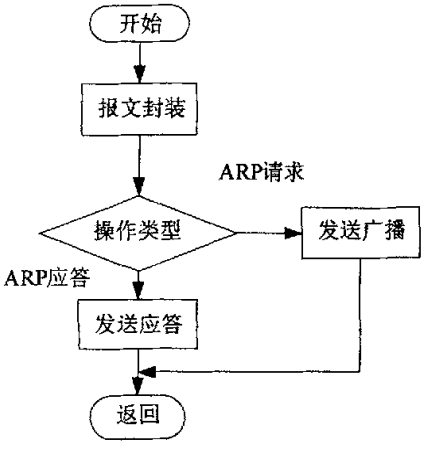

图3.17（a）ARP发送模块流程

Fig.3.17 (a) Flow chart of ARP sender module

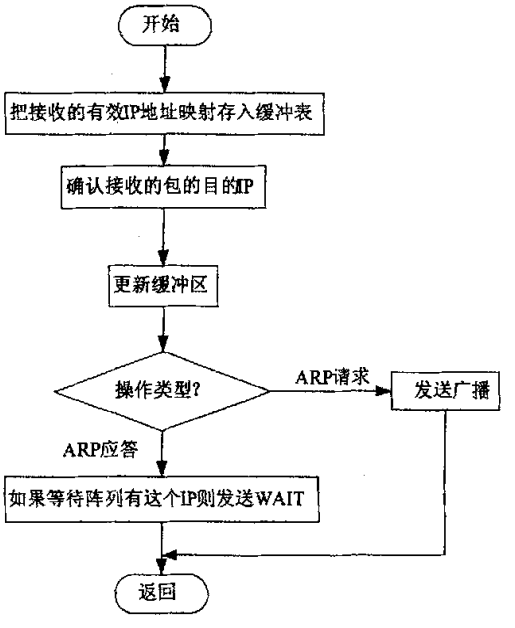

图3.17（b）ARP接收流程图

Fig.3.17 (b) Flow chart of ARP receiver module

②ARP地址解析

其中主要的流程图为ARP接收流程图和解析流程图，其关系如下：当从以太网传递来的帧确定类型为ARP时，ARP接收程序开始工作，负责判断是ARP请求还是ARP响应。这就完成了ARP协议的接收功能。而当发送IP包时，以太网首部中目的以太网地址需要

地址解析功能来填充目的以太网地址的选项，获得目的地址后就可以完成IP包的发送。如果没有得到目的物理地址，那么如何处理呢？在这个过程中设置一个等待阵列，待收到ARP应答后就可以把等待阵列的待发IP包传送出去。

等待阵列用C语言编程如下：

```
typedef struct
{
    UCHAR   xdata * buf;      //待发送的数据地址
    ULONG   ipaddr;           //发送方的IP地址
    UCHAR   proto_id;          //协议类型
    UINT    len;              //发送的数据长度
    UCHAR   timer;             //等待的时间，设置为两秒。
} WAIT;
```

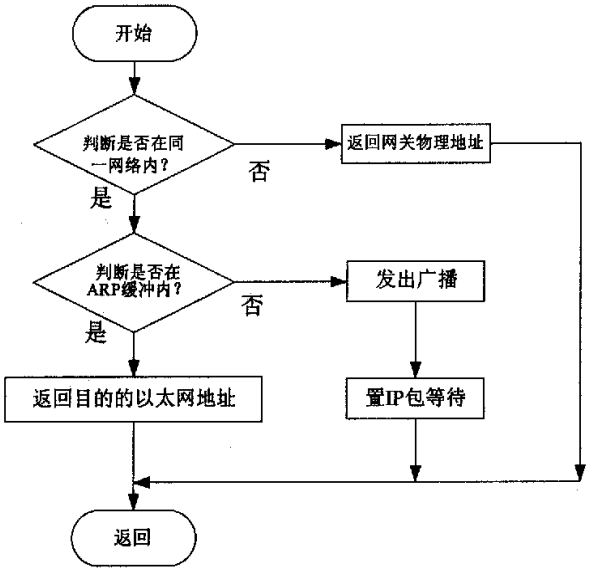

图3.18 ARP地址解析流程图

Fig.3.18 Flow chart of the resolution of ARP address

(2) IP 协议的实现

IP协议处在协议族中的第二层，在本系统中按标准编写了IP模块，主要处理网络层的数据的交换。IP协议是一种不可靠的无连接的数据报协议，不提供差错检验和跟踪。IP协议提供对于来自不同高层协议数据进行复用和分用。IP协议使网络之间的通信成为可能。如果嵌入式Internet需要跨越不同的网络进行通信就必须要实现IP协议。所以IP协

议需要完整实现。要特别注意IP包最大可以为65k字节，可以分段传输,而在嵌入式系统中根本无法容纳如此大的数据包,因此一般不支持分段。单片机一般采用发送小数据包的方式,以避免分段。IP层实现的是对要发送的消息打IP包，使之符合IP数据包的格式并把其送到物理层；对接收的来自物理层的协议进行IP解包，并且把数据送到上层传输（包括TCP和UDP层）。

IP协议的实现比较简单。 首先检查IP包的合法性。包括版本、checksum、目标地址等,然后根据协议类型提交到对应的协议处理中。与标准协议相比这里IP协议的实现进行了简化。主要有：

①只接受IPV4即32位的IP地址。

②只接受IP头长度为20字节的包。

③不允许IP包分段。这是因为系统不设置接收队列。只能处理一个包。

④不处理与ICMP、UDP、TCP无关的内容。

本设计主要使用两个模块来实现IP协议的功能。IP接收模块和IP发送模块。IP发送模块流程图如图3.19，IP接收模块流程图如图3.20。通过以下两个过程就可以完成IP协议的功能。

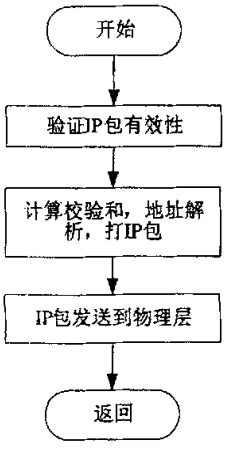

图3.19 IP发送模块流程图

Fig.3.19 Flow chart of IP sender module

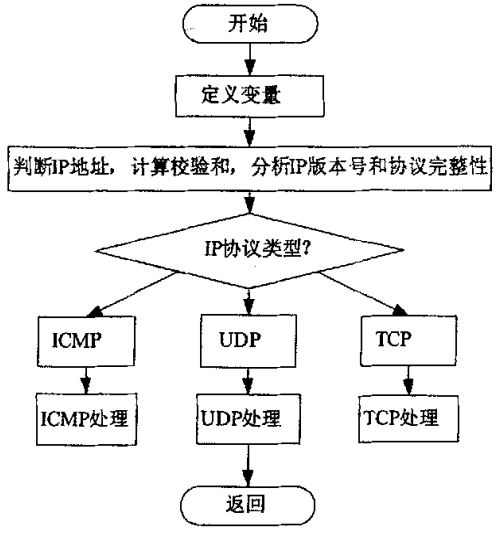

图3.20 IP接收模块流程图

Fig.3.20 Flow chart of IP receiver module

(3) 校验和算法的实现

大多数的TCP/IP协议中差错校验的方法是校验和的方法。校验和是在分组上的附加信息。在发送站先计算校验和，然后和分组一起发送出去，接收方对包括校验和的整个分组重复进行同样的计算。若正确就接受，否则就将其丢弃。发送端和接收端计算校验

和方法如下。

①发送端按以下方法计算校验和

将分组划分为N个部分，每个部分都为16位；用反码算术运算将这些分组进行累加；将最终的结果取反就得到校验和。

②接收端按以下方法计算校验和

将收到的分组划分为N个部分，每个部分都为16位；用反码算术运算将这些分组进行累加；将最终的结果取反，若结果为0则为正确，否则就为错误。

IP，ICMP，UDP，TCP在发送前就计算好校验和，以防止误传数据。其中IP只需要校验头部，ICMP是包含在IP数据中传输的，所以需要对ICMP整个报文进行校验和的计算。UDP校验和需要对三部分进行计算，UDP伪首部，UDP首部以及从应用层传来的数据。TCP与UDP类似。IP，ICMP，UDP，TCP计算校验和都采用checksum函数来进行计算[24]。

```
#Define UCHAR unsigned char
#Define UNIT unsigned int                          //简略定义
 //在以后的程序中都这个定义都有效。
unsigned int checksum(UCHAR xdata *check, UINT length)
                                                   //计算校验和
{
    LONG sum=0;
    UINT i;
    UINT xdata *ptr;
    ptr=(UINT xdata *)check;
    for (i=0;i<(length)/2;i++)
    {
        sum+=*ptr++;
    }
    if (length&0x01)                               //表示长度为单数
    {
        sum=sum+((*ptr)&0xff00);
    }
    sum=(sum&0xffff)+((sum>>16)&0xffff);           //高16位和低16位相加
    if(sum&0xffff0000)
```

```
    {                                          //表示有进位
        sum++;
    }
    return ( (UINT)((sum)&0xffff));
}
```

(4) UDP 协议的实现

在系统中主要是应用传输层的UDP协议来完成数据传输的，单片机将采集到的实时数据进行打包后，再在数据报头部加入UDP协议所需的控制字，就可以传输到网络层由IP模块在对UDP数据报进行相应的处理，处理后再发送到数据链路层，由数据链路层来完成数据的发送。它把数据分组从一台主机发送到另一台主机，但是并不能保证数据报能到达另一端，因此任何必需的可靠性必须由应用层来提供。

传输控制协议TCP和用户数据报协议UDP都是传输层协议。TCP协议提供面向连接的、端到端的可靠服务。在嵌入式系统中，可以不建立端到端的连接，只是将数据报送上网络，或者从网上接收数据报。而UDP协议提供无连接服务,它对要发送的数据不进行缓冲，直接把从应用层收到的数据加上UDP 头发送出去，能够保留各个消息之间的边界，并且UDP不会把应用层多次发送的数据合并成为一个包发送出去。这对于编写简单的请求响应模式的应用是很方便的,而且利用UDP协议对传输速率也没有太大的影响。UDP协议的数据报格式包括源端口、目的端口、消息长度、校验和。UDP模块主要包括两个模块，UDP发送模块和UDP接收模块。UDP发送、接收模块流程图如图3.21、3.22。

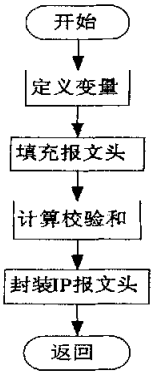

图3.21 UDP发送模块流程图

Fig.3.21 Flow chart of UDP sender module

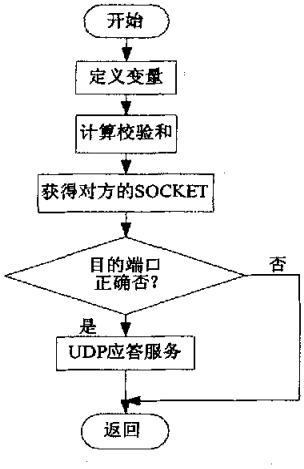

图3.22　UDP接收模块流程图

Fig.3.22 Flow chart of UDP receiver module

(5) TCP 协议的实现

TCP协议也是处在传输层中，本系统编写这一模块主要是为实现Web Server这一功能。建立Web Server后，客户端可以通过IE浏览器来查看装置的一些采集实时数据。TCP工作过程是：建立连接、数据传输、关闭连接。 TCP协议主要实现数据传输、可靠性、流控制、多路复用、面向连接等机制。

在将数据发向远方主机之前，必须先建立TCP接入。这意味着只有在两端都实现了TCP接入后，才可以进行端到端之间的数据交换。在建立TCP连接时，用到了三向握手机制。包含数据的每一个TCP段都应取得对方端口返回的应答段（ACK），作为握手信号来保证数据被可靠地接收。应答段本身不再需要应答，避免应答陷入无穷的嵌套。每一个TCP段中都包含一个序号，并以这个序号作为数据流的定位器，而返给客户机的应答号则表达所发来的数据已经妥收。

消除传输中的错误依赖持续跟踪已发出数据段的应答是否返回。在设定的时间段内，如果未收到该段的应答则应重发。如果还是未收到应答，则适当增加间隔时间再次重发。在总的极限时间段内一直不能等到应答返回，则本次接入失效不能再用，并应将出错情况及时通知应用程序。每次成功的接入，只要本地端口和对方端口没有执行过关闭操作，就一直是接通的。要想知道对方端口是否还处于接入状态，唯一的方法就是发数据进行探测。关闭 TCP 接入分为 4 步完成。

TCP实体所使用的基本协议是滑动窗口协议。每一个TCP段中，服务器都将其当前可用于接收的数据容量（即TCP数据窗，不计应答）通知对方，作为对方端口掌握发送的极限[25]。

由于单片机的资源有限，因此，在设计TCP协议时不得不采取大幅度的调整。同一时间只能有一个TCP任务，不支持分片和重组，只能同时接收和处理一个TCP包，不支持类型服务安全选项。

①TCP连接的建立

TCP连接的建立见下图3.23，为了建立一条TCP连接分以下几步。

第一步：请求端（通常称为客户）发送一个SYN段指明客户打算连接的服务器的端口，以及初始序号（ISN，在这个例子中为1415531521）。这个SYN段为报文段1。

第二步：服务器发回包含服务器的初始序号的SYN报文段（报文段2）作为应答。同时，将确认序号设置为客户的ISN加1以对客户的SYN报文段进行确认。一个SYN将占用一个序号。

第三步：客户必须将确认序号设置为服务器的ISN加1以对服务器的SYN报文段进行

确认（报文段3）。

这三个报文段完成连接的建立。这个过程也称为（three_way_handshake）三次握手。发送第一个SYN的一端将执行主动打开（active open）。接收这个SYN并发回下一个SYN的另一端将执行被动打开（passive open）。当一端为建立连接而发送它的SYN时，它为连接选择一个初始序号。ISN随时间而变化，因此每个连接都将具有不同的ISN。RFC 793[Postel 1981c]指出ISN可看作是一个32比特的计数器，每4ms加1。这样选择序号的目的在于防止在网络中被延迟的分组在以后又被传送，而导致某个连接的一方对它作错误的解释。

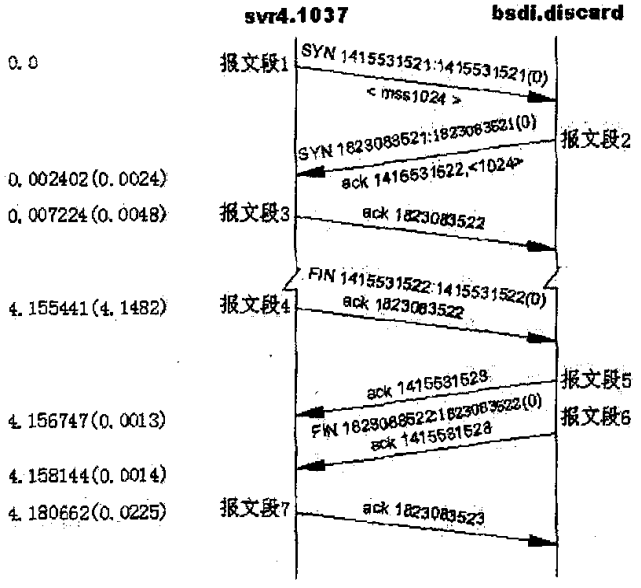

图3.23 TCP连接的建立

Fig.3.23 Establishing a connection of TCP

②TCP协议的终止

建立一个连接需要三次握手，而终止一个连接要经过4次握手。这由TCP的半关闭（half_close）造成的。既然一个TCP连接是全双工（即数据在两个方向上能同时传递），因此每个方向必须单独地进行关闭。这个原则就是当一方完成它的数据发送任务后就能发送一个FIN来终止这个方向连接。当一端收到一个FIN，它必须通知应用层另一端已经终止了那个方向的数据传送。发送FIN通常是应用层进行关闭的结果。收到一个FIN只意味着在这一方向上没有数据流动。一个TCP连接在收到一个FIN后仍能发送数据。而这对利用半关闭的应用来说是可能的，尽管在实际应用中只有很少的TCP应用程序这样做。首先进行关闭的一方（即发送第一个FIN）将执行主动关闭，而另一方（收到这个

FIN）执行被动关闭。通常一方完成主动关闭而另一方完成被动关闭。

图3.24显示了终止一个连接的典型握手顺序。本文省略了序号。在这个图中，发送FIN将导致应用程序关闭它们的连接，这些FIN的ACK是由TCP软件自动产生的。连接通常是由客户端发起的，这样第一个SYN从客户传到服务器。每一端都能主动关闭这个连接（即首先发送FIN）。然而，一般由客户端决定何时终止连接，因为客户进程通常由用户交互控制，用户会键入诸如"quit"一样的命令来终止进程。在图中，能改变上边的标识，将左方定为服务器，右方定为客户，一切仍将如显示的一样工作[26]。

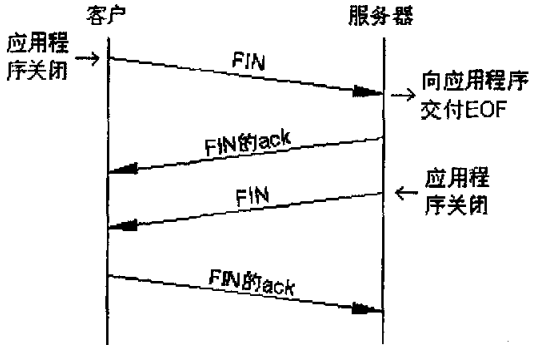

图3.24 TCP连接终止期间报文的发送

Fig.3.24 Closing connections of TCP

③TCP协议中复位的使用

终止一个连接正常的方式是一方发送FIN，这也称为有序释放，因为在所有的排队数据都已经发送之后才发送FIN，正常情况下没有数据丢失。发送一个复位的报文段也可以释放一个连接，这称为异常释放（abortive release）。异常终止一个连接对应用程序来说有两个优点：1）丢弃任何待发数据并立即发送复位报文段；2）RST的接收方会区分另一端执行的是异常关闭还是正常关闭。

RST报文中包含有个序号和确认序号。需要注意的是RST报文段将不会导致另一端产生任何响应。另一端根本不进行确认。收到RST报文的一方将终止连接，并通知应用程序连接复位。

④TCP状态机的实现

在TCP协议中最重要的是图3.16 所示的TCP状态变迁图，为了编程的方便，在这里使用C语言定义各个状态如下：

```
// TCP states
#define STATE_CLOSED        0    //关闭连接
#define STATE_LISTEN        1    //监听连接请求（被动打开）
```

```
#define STATE_SYN_RCVD      2    //已发送并接收SYN，等待ACK
#define STATE_ESTABLISHED   3    //建立连接，开始传输数据
#define STATE_CLOSE_WAIT    4    //收到FIN，并等待应用程序关闭
#define STATE_LAST_ACK      5    /收到的FIN已关闭，等待ACK
#define STATE_FIN_WAIT_1    6    //关闭，发送FIN，等待ACK和FIN
#define STATE_FIN_WAIT_2    7    //关闭，等待FIN
#define STATE_CLOSING       8    //同时关闭，等待ACK
#define STATE_TIME_WAIT     9    //主动关闭后2MSL等待状态
```

TCP有6个标志位，定义如下：

```
// TCP flag bits
#define FLG_FIN     0x0001
#define FLG_SYN     0x0002
#define FLG_RST     0x0004
#define FLG_PSH     0x0008
#define FLG_ACK     0x0010
#define FLG_URG     0x0020
```

TCP主模块的设计流程如下所示。

一个到达的TCP报文段、一个超时时间或者一个来自应用程序的报文，都可以调用主模块，这是一个非常复杂的模块，因为它要采取的动作取决于TCP的当前状态。有几种方法可以实现状态变迁图，包括为每一个状态使用一个进程，或者使用一个表，等等。但是为了实现方便，减少开销，我们使用一些情况来处理状态。本设计中仅仅使用了10个状态（其中的主动打开SYN_SEND状态对本服务器来说没什么意义）。每一种状态都按照状态变迁图来实现，下面是整个TCP状态变迁处理的流程[27]。

接收：TCP报文段，或超时事件。

查找TCP任务控制快中的STATE的值。

CASE（状态）：

CLOSED：服务器一直处于LISTEN状态，因此直接进入下一个状态LISTEN。

LISTEN：　若收到SYN报文（没有ACK）,则发送SYN+ACK报文,将状态改变SYN+RCVD状态。否则发送RST，重新同步对方主机。

SYN_RCVD：　若收到FIN报文则发送ACK报文,状态改变为CLOSE_WAIT状态，然后发送FIN报文，状态改为LAST_ACK。

若收到ACK报文,则确认对方的ACK报文后进STATE_ESTABLISHED。

ESTABLISHED：若收到FIN报文,则发送ACK，状态变为CLOSE_WAIT状态，

然后立即发送FIN和ACK报文，状态改为LAST_ACK

如果收到的是数据则发送ACK,然后将数据送到应用程序进行处理，打开保活定时

器。如果保活时间到了而没有数据流量，那么就发送ACK和FIN，然后将状态转变为STATE_FIN_WAIT_1。

CLOSE_WAIT：直接返回。

LAST_ACK：收到对方确认的ACK后，将状态改变为CLOSED状态。

STATE_FIN_WAIT_1：若收到FIN报文，发送ACK报文。如果对方ACK正确则进入TIME_WAIT，然后立即进入关闭状态；否则，并将状态改变为CLOSING。

若仅仅收到ACK报文,则将状态改变为FIN_WAIT_2。

STATE_FIN_WAIT_2：若收到FIN报文段则进入TIME_WAIT,并最后置为CLOSED状态。

STATE_TIME_WAIT：直接返回。

STATE_CLOSING:若收到ACK报文则进入TIME_WAIT,并最后置为CLOSED状态。

其他状态：直接返回。

⑤TCP差错控制机制

第一：大多数的TCP/IP协议都使用校验和机制来进行差错控制和校验。

第二：超时重传机制

TCP提供可靠的运输层。它使用的方法之一就是确认从另一端收到的数据。但数据和确认都有可能会丢失。TCP通过在发送时设置一个定时器来解决这种问题。如果当定时器溢出时还没有收到确认，它就重传该数据。对任何实现而言，关键之处就在于超时和重传的策略，即怎样决定超时间隔和如何确定重传的频率[28][29]。

本设计使用C建立一个结构体变量，来保存TCP处于连接时的信息。

```
typedef struct
{
  ULONG ipaddr;
  UINT port;
  ULONG his_sequence;
  ULONG my_sequence;
  ULONG old_sequence;
  ULONG his_ack;
  UCHAR timer;
  UCHAR inactivity;
  UCHAR state;
  char query;
```

```
} CONNECTION;
CONNECTION  xdata  conxn;              //定义了五个连接的信息
```

在重传TCP机制的模块中定义了时间，保证每次发送TCP数据后都有个计时机制，如主机发送完一个数据包，没有收到应答时每隔2S重新发送一次。把每个正处于STATE_ESTABLISHED状态的连接的信息保存在这个结构体数组中，对于一个正处于opening 或者 closing 状态的连接主机将发送一个RST报文。而对于一个处于ESTABLISHED状态的连接，检查ACK报文后，如果对方的ACK小于我方的序号，就重新传送，一般重传几次后仍然没有收到应答就发送复位报文。在本课题中设置重传次数最多为2次[30]。

对正处于连接的传输状态，每次TCP接收到有效数据的报文后，都打开TCP保活定时机制，每15秒检查连接，如果没有数据传递，那么连接自动断开。TCP保活函数就是完成这个功能。这个函数每隔0.5s就搜索CONNECTION的连接状态，如果保活时间到，那么它就关闭这个连接[31][32]。

3.4 小结

本章主要论述了网络仪表的结构和设计思想，主要从两个方面——硬件和软件进行详细的阐述。硬件以网络控制芯片RTL8019AS为核心，详细介绍了网络控制的硬件接口电路的设计；软件部分以TCP/IP协议为核心，详细阐述了TCP/IP协议的原理以及嵌入式系统中如何实现TCP/IP协议的问题，并给出了微型嵌入式TCP/IP协议的设计方法和软件程序流程图。本章是整个设计的核心部分，所给出的软件设计方法和电路设计都是很实用的，设计方案是经过实验验证过的，可以供相关设计人员参考使用。

第四章 系统软件设计

本章主要论述高炉煤粉喷吹状态监测装置的总体软件设计。软件设计包括上位机和下位机软件的设计，主要的实现系统中各个监测装置能够通过以太网同时向上位机发送实时采集的数据，满足最初的设计要求。下面将分两个部分进行说明。

4.1 软件设计总体思想

对与本系统来说，主要应用的软件是 C 语言[38]。特别是面向硬件的 C51，更适合本系统使用。每一种硬件都对应着一种 C 语言，但是它们核心结构和思想都是源自 ANSI C，只要掌握一种 C 语言，就触类旁通了。因此这里主要讲述的是面向单片机的 C51，有一定的代表性。而上位机软件形形色色，本文采用的是 C++ BUILDER，这是一款可以快速开发软件产品，并能够保证产品的性能以及享受到最新技术成果(如 COM 等)的开发工具。在众多快速应用程序开发环境工具中，C++ BUILDER 绝对是其中的佼佼者。在本设计中主要完成的是上位机监控画面，主要使用的是 C++BUILDER 的 WINSOCK 控件来完成通信和控制功能，也实现了故障诊断功能。

4.2 下位机程序设计

由于本设计的工作量十分庞大，为了编程的方便，笔者主要使用了 10 个模块，分别是 ANALOG、ARP、CHECKSUM、HTTP、TCP、UDP、ICMP、IP、TIMER、以及 ETH 模块设计。

其中 TCP/IP 协议的模块在前一章已经讲述过，这里就省略了。ANALOG 模块主要负责完成对传感器输出信号采集的功能；TIMER 模块主要完成实时时钟的功能；ETH 模块主要完成 RTL8019AS 以太网控制芯片的初始化和控制。上述这几个模块都在主函数里进行调用，它们的关系是等同的，仅仅是分时操作而已。下面先介绍主程序。

4.2.1 主程序 MAIN

主程序主要是完成系统参数的初始化和实时调用任务。在 MAIN 中主要完成了单片机寄存器和端口初始化，内存空间的合理分配，网络参数初始化以及由一个 while 循环而进行的任务调度[31][32]。

主要流程图如下：

图 4.1 中当收到报文的标志有效后，应立即清除标志位，等待下次标志的有效。每次任务的调度都执行相关的模块，当网卡收到数据包标志有效，执行以太网模块；当

ARP 缓冲更新标志或者 ARP 重传标志有效，执行 ARP 模块；当 TCP 重传标志或者 TCP 状态保活标志有效时执行 TCP 模块；当模拟输入标志有效时执行 ANALOG 模块。

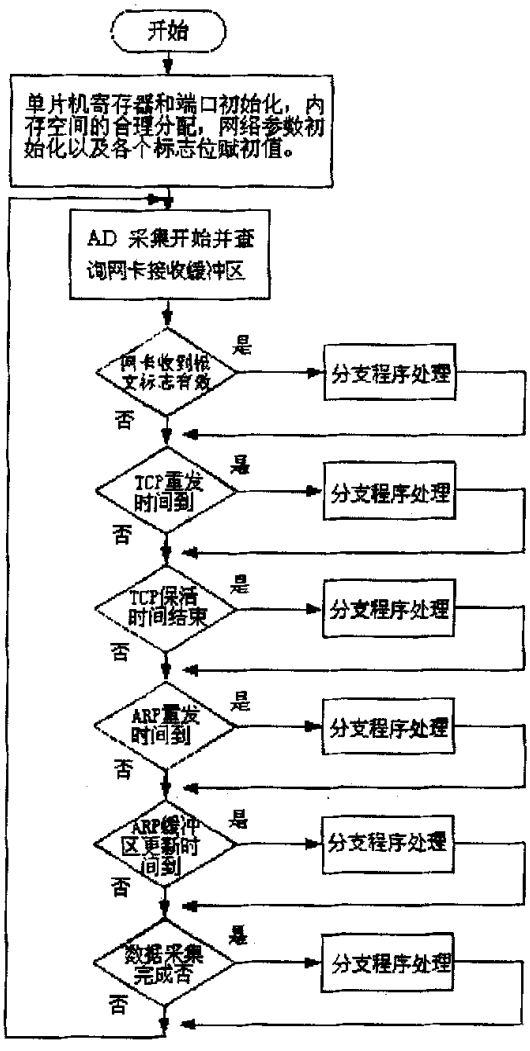

图 4.1 主程序流程图

Fig .4.1 Flow chart of main program

其中标志位用 C 语言设置如下：

```
#define EVENT_ETH_ARRIVED        0x0001
                                          //网卡收到数据包标志
#define EVENT_AGE_ARP_CACHE      0x0002
                                          //ARP 缓冲更新标志
#define EVENT_TCP_RETRANSMIT     0x0004
```

```
                                                    //TCP 重传标志
#define EVENT_TCP_INACTIVITY        0x0008
                                                    //TCP 状态保活标志
#define EVENT_ARP_RETRANSMIT        0x0010
                                                    //ARP 重传标志
#define EVENT_READ_ANALOG           0x0020
                                                    //模拟输入标志
```

以上程序中最重要的是 RTOS 即实时控制时间函数，下面就介绍 TIMER 模块。

4.2.2 TIMER 模块

实时时钟对定时器要求十分苛刻，本文使用了 C8051F021 优先级比较高的定时器组成实时控制系统的时钟。

首先使用定时器 2，每 25ms 中断一次，使用 16 位自动装载方式。

由于程序比较简单，这里给出中断服务源程序。

```
void timer2_interrupt (void) interrupt 5
{
    static UINT count1 = 0;
    static UINT count2 = 0;

    TF2 = 0;                // 清除中断标志
    count1++;
                    // 这些时间每 0.5s 发生一次，但是分时操作，相互不影响
    if (count1 == 5) event_word |= EVENT_ARP_RETRANSMIT;

    if (count1 == 10) event_word |= EVENT_TCP_INACTIVITY;

    if (count1 == 15) event_word |= EVENT_READ_ANALOG;

    if (count1 == 20)
    {   count1 = 0;
        event_word |= EVENT_TCP_RETRANSMIT;
    }

    count2++;
    if (count2 == 2401)
    {
```

```
                // 这个事件每 60.025 seconds 执行一次，全部任务结束。
            count2 = 0;
            event_word |= EVENT_AGE_ARP_CACHE;
        }
    }
```

这个程序主要完成的是分时处理任务。其它的对 ROM 和存储器的操作比较简单，由于篇幅有限，这里就不详细说明了。

4.2.3 ANALOG 模块设计

该模块主要完成对煤粉喷吹状态监测装置的温度、浓度、速度信号进行采集。

主要包括寄存器初始化和 AD 采集，在硬件系统中这是比较常见的。

由于在本系统中，已经使用的中断比较多，不好处理，因此应尽量避免使用中断来调用 AD 转换。本文采用写转换控制字(busy)方式来调用 AD 转换，在 TIMER 模块中已经对 AD 做了处理，每 0.5S 调用一次，这样既保证了采样的精度，也避免了操作上的麻烦。图 4.3 为 AD 采集程序流程图。

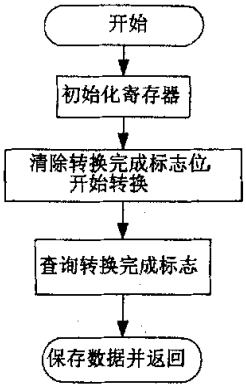

图 4.2 AD 采集模块程序流程图

Fig.4.2 Flow chart of AD module

4.2.4 ETH 模块设计

该模块主要解决数据在链路层传输的问题。主要是对网络层的数据报进行处理，发送的数据通过该模块写到 RTL8019AS 的发送寄存器中，同时从 RTL8019AS 接收寄存器中读取接收到的数据。

RTL8019AS 芯片在第三章里已有详细的说明，这里就软件编程方面的问题探讨一下。对 RTL8019AS 进行操作，需要对其内部结构十分了解，首先对其存储器结构作个说明，然后给出软件设计的流程图。

(1) RTL8019AS 网卡 DMA 和内部 RAM

RTL8019AS 网卡内部有两个 RAM 区，一块为 16K 字节，地址为 0x4000H-0x7FFFH；另外一块为 32 字节，地址为 0x0000H-0x001FH；RAM 按照页来存储，每 256 字节为一页，本文将 RAM 的前 12 页作为发送缓冲区（即 0x4000H-0x4BFFH），后 52 页（即 0x4C00H-0x7FFFH）存储区作为接收缓冲区，第 0 页叫 PROM，只有 32 个字节，地址为 0x0000H-0x001FH，用于存储以太网的物理地址。

RTL8019AS 内部可分为远程 DMA 和本地 DMA，MAC 逻辑、数据编码解码逻辑和其它端口。MAC 逻辑主要完成以下功能：当单片机向网上发送数据时，先将一帧数据通过远程 DMA 通道发送到 RTL8019AS 的发送缓冲区内，然后发出发送命令；当 RTL8019AS 完成了上一帧数据发送后，再开始此帧的发送。RTL8019AS 接收到的数据要通过 MAC 比较，CRC 校验后，有 FIFO 存到接收缓冲区：当收满一帧时，以中断或者寄存器标志的方式通知主处理器，FIFO 逻辑对收发数据作 16 字节的缓冲，以减少对本地 DMA 请求的频率。

要接收或发送数据必须通过 DMA 读写 RTL8019AS 内部的 16K 的 RAM，它实际上是双端口的 RAM，有两套总线连接到此 RAM，一套为本地 DMA，另外一套为远程 DMA[46]。

本设计使用 0x40-0x4B 为网卡的发送缓冲区，共 12 页，刚好可以存储 2 个最大的以太网包。使用 0x4C－0x7F 为网卡的接收缓冲区，共 52 页。设置接收缓冲区页码有两个寄存器决定：

Pstart(page start register)和 Pstop(page stop register)

A15	A14	A13	A12	A11	A10	A9	A8

这两个寄存器都是 16 位 RAM 地址的高 8 位，也就是页码。因此 PSTART=0x4C,PSTOP=0x80(0x80 为停止页，就是直到 0x7F，是接收缓冲区，不包括 0x80)。

同理设置发送寄存器 TPSR=0x40，表示发送缓冲区从页 0x40 开始，0x40-0x4B 共 12 页作为发送缓冲区。一个最大的数据包需要 1514 字节+4 个字节的校验，刚好需要 6 页（256*6=1536 字节）。因此 12 页正好能存储两个最大的以太网包。设置两个发送缓冲区是有目的的，这里假设前一个 6 页的发送缓冲区为 SEND1，后一个为 SEND2。用户在使用过程中，将数据包放在 SEND1，然后启动发送。发送过程中，如果用户还有数据要发的话，那么就可以把要发送的数据包放在 SEND2 中，等到 SEND1 的数据发送完毕后，就可以马上发送 SEND2 的数据，这样就实现了连续发送数据包。

设置了接收缓冲区后，如何确定接收到的第一个数据包放在哪里，这由寄存器CURR 决定。控制接收缓冲区有两个寄存器 CURR 和 BNRY。CURR 是网卡写内存的指针。它指向当前正在写的页的下一页。BNRY 是读指针，指向用户已经读走的页。那么初始化它就应该指向 0x4C+1=0x4D 。网卡写完接收缓冲区一页，就将这个页地址加1，CURR=CURR+1。这是网卡自动加的。当加到最后的空页(0x80,PSTOP)时，将 CURR 置为接收缓冲区的第一页（这里是 0x4C，PSTART），也是网卡自动完成的。当 CURR =BNRY 时，表示缓冲区全部被存满，数据没有被用户读走，这时网卡将停止往内存写数据，新收到的数据包将被丢弃不要，而不覆盖旧的数据。此时实际上出现了内存溢出。而 BNRR 要由用户来操作。用户从网卡寄存器读走一页数据，要将 BNRY 加 1，指向已经读走的页，然后再写到 BNRY 寄存器。当 BNRY 加到最后的空页（0x80，PSTOP）时，同样要将 BNRY 变成第一个接收页 PSTART， BNRY=0x4C; CURR 和 BNRY 主要用来控制缓冲区的存取过程，保证能顺次写入和读出。当 CURR=BNRY+1（或当 BNRY =0x7F,CURR=0x4C)时，网卡的接收缓冲区里没有数据，表示没有收到数据包。 用户通过这个判断知道没有包可以读。当上述条件不成立时，表示接收到新的数据包。然后用户应该读取数据包，直到上述条件成立时，表示数据包已经读完，此时停止读取数据包。

对于单片机来说，要操作网卡必须通过远程 DMA。远程 DMA 操作如下。RSAR0 和 RSAR1 这两个寄存器是远端 DMA 发送首地址，RBCR0，RBCR1 这两个寄存器是远端 DMA 发送数据的字节长度。CRDA0 和 CRDA1 是指向远端 DMA 的地址，指向当前正在读写的数据。主机设置好远端 DMA 开始地址和远端 DMA 数据字节数，并在命令寄存器 CR 中设置读/写(reg00=0x0A/reg00=0x12)命令,就可以从远端 DAM 寄存器中读出或者写入芯片里的 RAM。简单的说就如同前文说的是两套总线，但是操作的地址是一样的。远程 DMA 是在主控制器参与下完成的，即主机给出起始地址和数据长度就可以对芯片的内部 RAM 进行读写操作。

(2) RTL8019AS 控制程序设计

RTL8019AS 控制程序的目的有两个，分别是发送和接收 IP 包、发送和接收 ARP 包。在发送时，要把待发送的数据包按指定的格式写入芯片的 FIFO，并启动发送命令，那么 RTL8019AS 会自动的把数据包转换为以太网物理帧格式进行发送。反之 RTL8019AS 收到数据包以后，它会把数据还原成数据，按指定的格式放到 FIFO 中。本模块主要包括四个函数，分别是 init_8019( )； query_8019( )； send_frame( )； rcve_frame( )。各个函数功能及流程图如下。

①init_8019 函数

主要完成 RTL8019AS 工作参数的初始化。

各个寄存器在第三章已有说明，详细资料请参照 RTL8019AS 的 DATASHEET。

这里给出简单的程序，用 C51 编写。

```
void init_8019 (void)
{
reg00=0x21;      //选择页 0 的寄存器，网卡停止运行，因为还没有初始化。
reg01=0x4c;      //寄存器 Pstart 设置接收缓冲区起始地址
reg02=0x80;      //Pstop 设置接收缓冲区结束地址
reg03=0x4c;      //设置 BNRY=0x4c
reg04=0x45;      //设置 TPSR=0x45
reg0c=0xcc;      //设置 RCR=0xcc
reg0d=0xe0;      //设置 TCR=0xe0
reg0e=0xc8;      //DCR 数据配置寄存器 8 位数据 dma
reg0f=0x00;      //IMR disables all interrupt
page(1);         //选择页 1 的寄存器
reg07=0x4d;      //CURR 设置写指针
reg08=0x00;      //MAR0
reg09=0x41;      //MAR1
reg0a=0x00;      //MAR2
reg0b=0x80;      //MAR3
reg0c=0x00;      //MAR4
reg0d=0x00;      //MAR5
reg0e=0x00;      //MAR6
reg0f=0x00;      /MAR7
reg00=0x22;    //选择页 0 寄存器，网卡执行命令。
}
```

②query_8019 函数

主要是完成对 RTL8019AS 的查询，看是否有新的数据从上位机传输到下位机。如果有新的数据到来就触发 EVENT_ETH_ARRIVED 事件。

```
void query_8019 (void)

{
char bnry，curr;
page(0);
bnry=reg03;          //bnry page have read 读页指针
page(1);
```

```
curr=reg07;          //curr writepoint 8019 写页指针
page(0);
if ((curr==0))    return;
bnry=bnry++;
if (bnry>0x7f)   bnry=0x4c;
if (bnry! = curr)       //此时表示有新的数据包在缓冲区里
{
EA = 0;
event_word |= EVENT_ETH_ARRIVED;
EA = 1;
}
reg0b=0x00;   reg0a=0x00;   reg00=0x22;    //complete dma page 0
}
```

③send_frame 函数

send_frame 函数主要完成发送数据包的程序。先将待发送的数据包存入芯片 RAM，给出发送缓冲区首地址和数据包长度(写入 TPSR、TBCR0,1)，启动发送命令(CR=0x3E)即可实现 RTL8019AS 发送功能。RTL8019AS 会自动按以太网协议完成发送并将结果写入状态寄存器。RTL8019AS 发送包帧结构如表 4.1 所示。

表 4.1 8019 发送包帧结构

Table 4.1 Format of sending datagram frame of 8019

字段	目的 MAC 地址	源 MAC 地址	类型 TYPE/ 长度 LEN	数据域 DATA	填充 PAD
位	48	48	16	<=1500 字节	可选

发送一个数据包，长度最小为 60 字节，不足需要填充，最大为 1514 字节。知道了发送的格式，那么很简单就可以完成数据的发送。由于本课题中选择 12 页作为发送缓冲区，这里我们可以同时使用两个发送缓冲区。意思很简单，先填充发送缓冲区 1，并设置为发送；同时可以填充另外一个发送缓冲区，等待发送缓冲区 1 发送完毕立即就可以发送另外一个缓冲区的数据了。这样就提高了的效率，实现了发送数据的连贯性。

发送的过程分以下几步：

第一：首先是装帧，就是按照 TCP/IP 协议的格式将数据封装。这个过程很简单就是把协议按照格式封装，IP 帧加上 IP 头文件，然后按照以太网的帧格式进行封装；同样 ARP 报文也按照以太网帧格式进行封装。

第二：将帧送入 NIC 的发送缓冲区，这个过程其实就是所谓的远程 DMA 操作。在前文已有说明。启动远程 DMA 写操作时设置命令寄存器 CR 为 0x12H，对 NIC 操作完成后，远程 DMA 就从数据 I/O 端口直接取走数据，这个过程是由 8019 自动完成。远程 DMA 最多发送 6 次，这在程序使用中要注意。执行完远程 DMA 操作后，就可以使用本地 DMA 进行发送数据了。

第三：启动本地 DMA，把数据发送到以太网。这个过程同远程 DMA 很类似，也是对三个寄存器进行操作。设置发送页起始地址寄存器 TPSR，设置发送字节计数寄存器 TBCR0 和 TBCR1，然后设置命令寄存器 CR 为 26H，这样就完成了本地 DMA 操作。

④Rcve_frame 函数

Rcve_frame 函数主要完成数据帧的接收，本设计主要是指从以太网上接收数据帧存于 NIC 接收缓冲区内，然后进行类型和地址判断，分别加以处理。整个过程分为三个阶段：首先帧接收，由 NIC 通过本地 DMA 从网络上接收帧，并存入接收缓冲区；帧读入，通过远程 DMA，将数据从接收缓冲区内的数据读到 MCU 内。最后按照不同帧格式识别不同的协议，分别进行处理。RTL8019AS 接收包帧结构如表 4.2 所示：

表 4.2 8019 接收包帧结构

Table 4.2 Format of receiving datagram frame of 8019

字段	接收状态	下一页指针	以太网帧长度	目的 MAC 地址	源 MAC 地址	类型/长度	数据域	填充	校验
位	8	8	16	48	48	16	<1500 字节	选	32

帧接收：

这个过程是由 NIC 通过本地 DMA 自动完成的，并不是开发程序所能控制的，因此只要在初始化时完成对与接收帧有关的寄存器进行适当的操作就可以了。这里要对 PSTART，TSTOP 寄存器进行初始化来构造接收缓冲区，页地址为 0x4CH-0x7FH；同时把 CURR 初始化为本地 DMA 的写指针；把 BNRY 初始化为远程 DMA 的读指针。

当网络上有帧到达时，NIC 开始在 CURR 寄存器指定的位置上存放数据，如果接收成功，RTL8019AS 会在存放数据前留出四个字节的空间来存储和接受与帧有关的状态和信息。

帧读入：

它是由远程 DMA 读操作和 MCU 读数据操作配合完成的。为了使远程 DMA 进行读操作，必须初始化相应的寄存器，然后启动远程 DMA 读操作和 MCU 读数据操作，就可以完成帧读入的工作。为了维护缓冲区，在 NIC 初始化时可以设置一个 nextpage 变量指示要读的下一个地址，而 nextpage 变量取值为下一个要读的地址值减 1，每次读如前都把它设置为 BNRY 的当前值；当确认从网络接收数据后，则改变它的值为 RTL8019AS 接收包帧结构表中所列的下一页指针的值减一，然后把它赋值给当前的 BURY 指针即表示当前已经读到这个位置。

与远程 DMA 相关的寄存器有远程开始地址寄存器 RASR；远程字节计数器 RBCR，字节的信息可以从 RTL8019AS 接收包帧结构中的以太网的长度字段中获得。使用中须注意当接收的数据为空时，在 C 语言里表示为 NULL，则在使用中跳过这个接收区域，在新的空白页中重新开始计数即设置 BNRY=CURR-1。

帧处理：

主要是将 MCU 得到的数据进行分析，如果是 IP 包则送往 IP 模块进行处理；如果是 ARP 包则送往 ARP 模块进行处理。

以上便完成了 ETH 模块的主要设计。

4.2.5 WEB 模块设计

目前企业中的计算机网络绝大多数仅局限于管理信息系统的应用。另一方面，许多车间装置安装了各种工业测控系统，反映现场运行于工况的画面、参数和报表，但大部分只能在本地控制室内进行，只有现场操作人员能了解，无法被生产的高层决策者直接了解和应用。随着 Intranet 在企业内部的应用，可以利用企业信息网络，进行工厂实时运行数据的发布和显示。管理者通过 Web 浏览器对现场工况进行实时的远程监控、远程设备调试和远程设备故障诊断和处理[26][27][28]。

本文采用 HTML 语言来编写 Web 程序，来实现数据的接收与发送。Web Server 采用的是 TCP 协议，占用的端口是 80。图 4.4 给出了 web 客户-服务结构示意图。

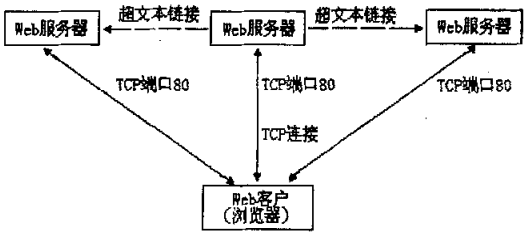

图 4.4 web 客户-服务结构示意图

Fig.4.4 The structure of web client server

Web 模块工作过程如下：首先对其进行初始化，监听 HTTP 的端口 80。而后根据远端主机是否与本机相连，判断是否接收新数据；根据远端主机是否给本机发确认，判断远端主机是否接收到本机的数据。

客户端可以通过 IE 浏览器来查看装置的一些采集实时数据，如图 4.5。

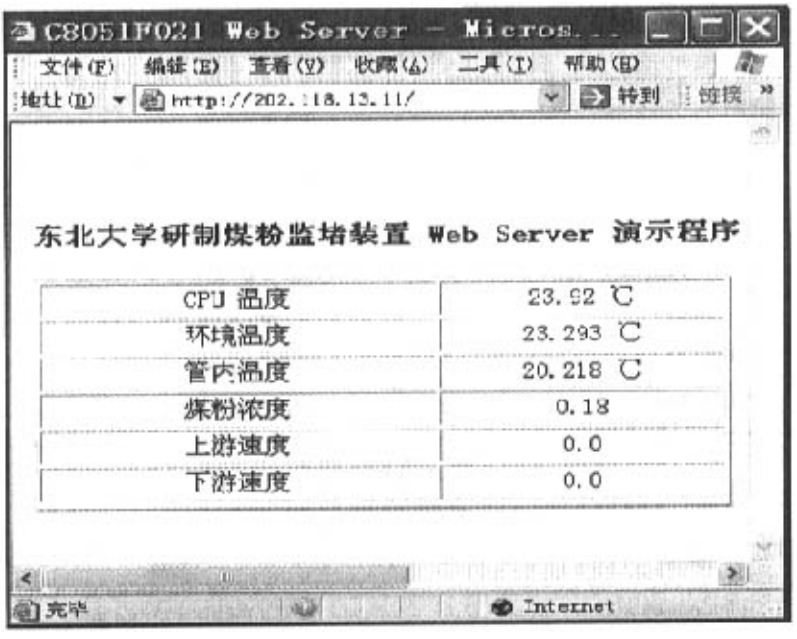

图 4.5 Web Server 演示图

Fig.4.5 The picture of Web Server demo

4.3 上位机程序设计

实现了单片机的 TCP/IP 协议的嵌入以后，就可以把单片机采集的数据通过以太网进行传输。因此，设计中必须在 PC 机上进行 WINDOWS 网络编程，依靠 PC 机强大的处理能力对单片机传来的数据进行处理和监控。这里将分两部分进行说明，主要包括通信程序和监控程序。本文利用 C++builder 编写了煤粉喷吹系统状态监测软件，该软件可以显示系统上传的状态数据，如两支管的温度、浓度、速度，还可以显示系统参数设置、温度浓度变化曲线等状态信息，而且根据专家系统中的故障诊断原理，实现了对煤粉喷吹状态监测系统的故障诊断[33][34]。下面就主要介绍一下该软件的编制。

4.3.1 WinSock 控件介绍

在设计中，采用的是C++Builder[37]自带的Winsock控件，用它可以与远程计算机建立连接, 并通过使用者数据记录通信协议(UDP)或者传输控制协议(TCP)来进行数据交

换。两种协议各有优点，前者传输数据快而后者数据错误较少，在一些数据比较重要时采用TCP协议传输。

Winsock控件的属性如表4.3。只要在控件相关事件函数里添加处理事件程序就可以了。当系统触发相应的事件后，就执行相关的程序。表4.4给出了事件选项。

表4.3 Winsock控件的属性

Table 4.3 Winsock Control properties

属性	描述
Active	Bool类型，用来激活一个Socket
RemoteHost	传回远程计算机的IP地址
RemotePort	传回或设置要连接的远程端口号
Name	Winsock控件名称
LocalHost	本地计算机的IP地址
LocalPort	本地计算机的连接端口号

表4.4 Events 选项

Table 4.4 Events

事件	描述
OnConnect	连接成功时触发该事件
OnCreateHandle	创建一个新的连接
OnDestroyHandle	删除一个连接
OnDisconnect	断开全部连接
OnError	当连接发生错误时，触发它
OnReceive	处理接收数据事件
OnSend	处理发送数据事件

4.3.2 煤粉喷吹系统状态监测软件介绍

本软件主要是完成显示系统上传的状态数据，如各个支管的温度、浓度、速度，提供系统参数设置、各支管温度浓度当前量、各支管趋势量等状态信息的查询，而且根据专家系统中的故障诊断原理，实现了对煤粉喷吹状态监测系统的故障诊断。图 4.6 是该软件的主界面，运行主界面后，就可以显示各个支管的状态信息，从程序的运行可知，该系统在实验室里实现了两个支管同时通信的目标。图中的高炉圈图可以显示系统是否处于正常喷吹状态，当某支管堵塞时，相对的圆圈显示红色。这样就可推而广之，当系

统由 38 个支管组成时，在主界面就可以同时察看各个支管的信息了。

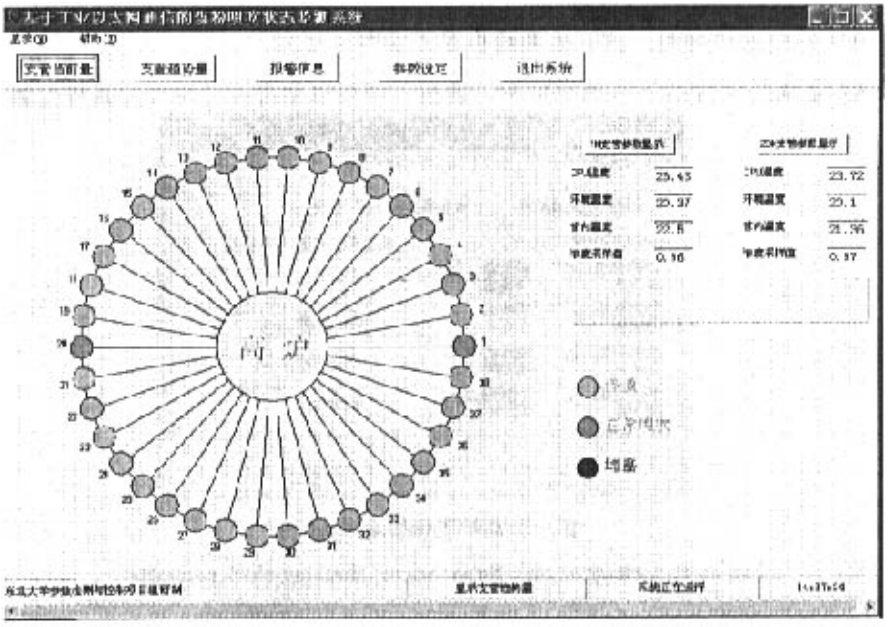

图 4.6 程序主界面

Fig .4.6 The main interface of the software

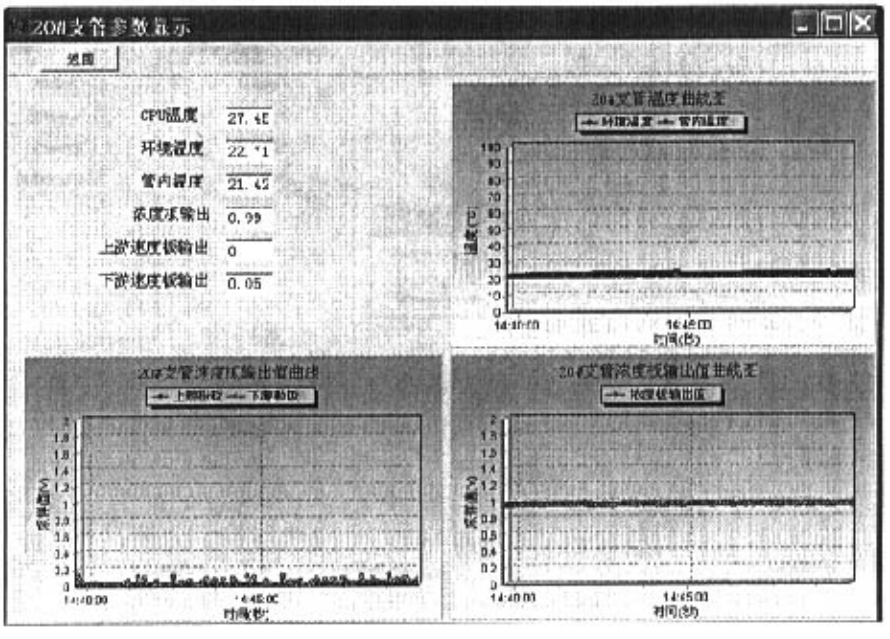

图 4.7 20#支管参数显示图

Fig.4.7 The parameter of 20# pipe display

图 4.7 是 20#支管参数显示图，在该图中实时地显示 20#支管的各方面状态信息，包

括 CPU 温度、管内温度、环境温度、浓度板输出、上下游速度；同时还显示了温度、浓度、速度的变化趋势图，这五条曲线也是在实时变化的。

图 4.8 是高炉支管检堵参数设定图，该窗口中可以全部或分别设定各种报警信息。

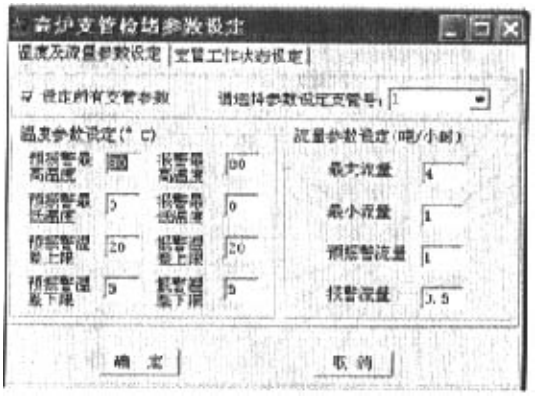

图 4.8 高炉支管检堵参数设定

Fig4.8 The setting of blast furnace pipes checking block parameter

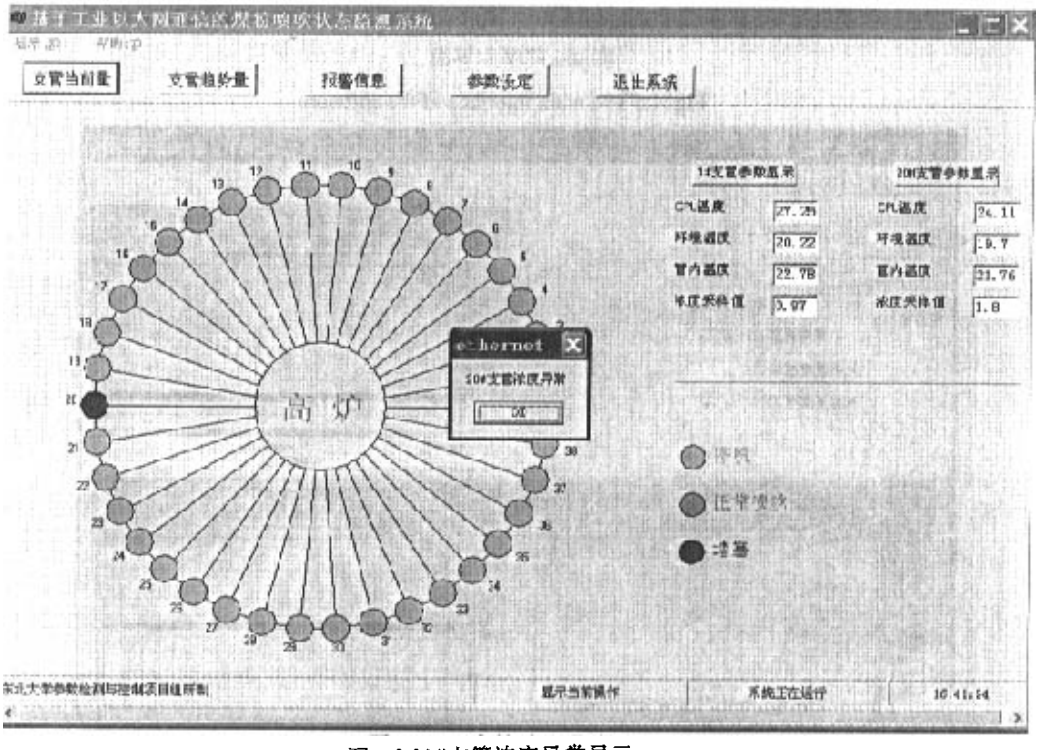

图 4.9 20#支管浓度异常显示

Fig.4.9 20# pipe concentration abnormal display

图 4.9 是 20#支管浓度异常时显示图，从图中可以看出，当浓度超过设定值时，相对应号码的圆圈显示红色，并出现报警信息。

图 4.10 是报警信息的查询界面。记录着系统出现堵塞的信息，以供查询。

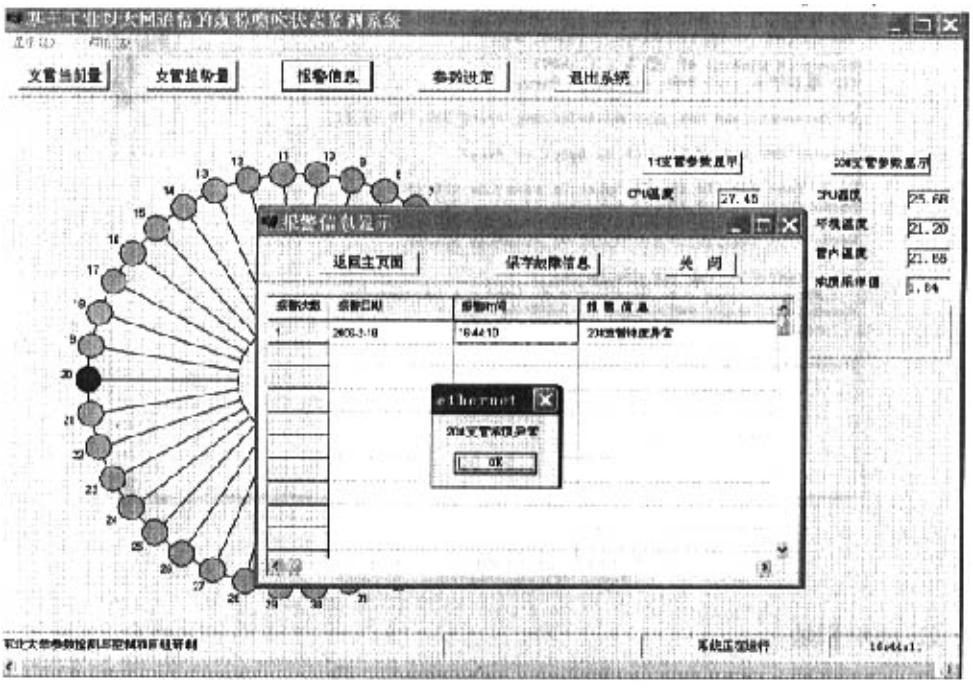

图 4.10 报警信息显示图

Fig.4.10 The information of warning

4.4 系统测试

在上面几节着重介绍了系统软件的总体编程思想和具体的每个模块的功能，这一部分主要介绍一下整体的软件系统的运行情况。各个模块在 Cygnal 开发环境下编译完成后，通过 JTAG 端口下载到单片机的 Flash 中。这样，编写的 TCP/IP 协议栈、数据采集子程序和 WEB Server 程序就固化在单片机中了。以下分别看看协议是否能够编写成功，是否能达到预期的设想。

4.4.1 通信程序部分

这部分主要编写了 ARP、ICMP、TCP、UDP、HTTP、WebServer 协议程序，下面看看各个模块的运行状况。

(1) ICMP 模块

这一模块的作用主要是上位机通过 ping 这一命令来判定下位机是否与上位机连接上。在现场调试时，应用这一模块能够较快地判断出系统装置是否已经连接到工业以太网中。实现的步骤为：在 pc 机运行栏中数据 ping 202.118.13.11(设定的下位机的 IP 地址)，

运行结果如图 4.9，从图中就可以判定下位机已经与上位机连接上了。

图 4.9 ping 命令的实现

Fig.4.9 The realization of ping

(2) UDP 模块

模块是用来判定系统通信是否正常，可以采用 C++Builder 中的 UDP 控件编写一个简单的应用程序。在下位机的 UDP 模块中已经添加了一个反馈函数，这一函数的作用是当 PC 机向单片机发送数据后，单片机接收数据的同时再把数据反馈给 PC 机。

图 4.10 给出了模块运行结果。

图 4.10 UDP 模块运行结果

Fig.4.10 Result of UDP module function

其他模块的运行结果已经在上一章给出了，这里不再讨论了。系统可以通过 TCP/IP 协议通信，把采集来的数据发送到上位机中。因为在协议中加入了 CRC 校验，当程序发现数据的校验位发生错误后，网卡接口芯片就会自动将这一次的数据丢弃，不会把有错误的数据发送到上位机。这样，可以确保上位机接收到的数据的正确性，确保监控系

统对煤粉喷吹状态的判定。

4.4.2 监控程序部分

这部分软件主要作用是将下位机发送上的实时数据显示到 PC 机上，并根据故障诊断的专家规则对系统的实时的故障做出判断。显示的数据包括有各个支管的管内温度、环境温度和通过支管的质量流量。在 4.3 节中已经作了详细的介绍，可知这部分的功能可以实现。

4.5 小结

本章主要论述高炉煤粉喷吹状态监测装置的总体软件设计。其中包括上位机和下位机软件的设计，主要的实现系统中各个支管能够通过以太网同时向上位机发送实时采集的数据，满足系统的实时性要求。通过多次实验验证，可以得出编写的 TCP/IP 协议能够满足系统实时性的要求，实现了通过交换机的双机通信，即两台下位机通过交换机同时向上位机发送数据，两台下位机互相不受对方的影响；而且，上位机能同时接到它们发送的实时数据，并准确的显示在监控界面上。可以满足系统在实时性和准确性上的要求。

第五章 结 论

工业以太网近几年得到广泛重视和深入研究，并在工控领域有了许多成功应用的例子。将工业以太网技术用于煤粉喷吹状态监测系统是一个非常有价值的研究课题。本文是以高炉煤粉喷吹状态监测系统为研究对象，对以太网芯片接口的设计以及对芯片寄存器的操作都作了十分详细的说明，并通过系统的硬件和软件的设计与编制完成了对煤粉喷吹状态数据的实时采集和处理，实现了远程监控的基本目标。完成的主要工作有：

（1）对工业以太网控制系统做了深入的研究，设计完成了以 CYGNAL C8051 F021 微控制器和 RTL8019AS 网卡接口控制芯片为核心的以太网测控网关，开发出了用于工业以太网控制系统中的现场测控设备；

（2）编写嵌入式 TCP/IP 网络协议，实现了通过以太网同时传送各个支管状态信息的目的；并且编写了 Web Server 功能，实现利用 IE 浏览器传递数据的目的。

（3）利用 C++builder 编写了煤粉喷吹系统状态监测软件，该软件可以显示系统上传的状态数据，如两支管的温度、浓度、速度，还可以显示系统参数设置、温度浓度变化曲线等状态信息，而且根据专家系统中的故障诊断原理，实现了对煤粉喷吹状态监测系统的故障诊断。

实验结果表明，利用以太网通信的煤粉喷吹状态监测系统实现了实时的传输数据，即同时获得各个支管的状态信息，改变了以往现场总线巡检方式获取信息，实现了数据的实时传送，避免了状态数据的丢失；能够实现利用 IE 浏览器传递数据，完成远程监视、控制、诊断的功能。研究成果对今后的工业以太网现场仪表的设计与开发具有一定的参考价值。由于时间和条件的限制，本设计还有许多地方有待于完善。比如可以采用更高端的 MCU，如 arm 单片机、DSP 等，这样可以增强单片机的运算速度；还可以采用更快的网卡接口芯片如 100M 网卡芯片，加快数据的传输速度。从而达到增加系统实时性的目标。

参考文献

1. 冯冬芹，金建祥，褚健. 浅谈以太网应用于工业现场的关键技术[J]，世界仪表与自动化，2002(4)
2. 成继勋. 广义现场总线标准与工业以太网[J]，电气自动化，2002，(4)：8-11
3. W.Richard Stevens. TCP-IP 详解卷 1：协议[M], 1995
4. 曾家智等. 计算机网络[M]，电子科技大学出版社，2002
5. 郑宗汉. 实时系统软件基础[M]，清华大学出版社，2003
6. 陈积明. 工业以太网的研究现状及展望[J]，化工自动化仪表，2001，28(6)
7. 宋真君. 基于工业以太网的分布式控制系统的通信研究[J]，厦门大学学报，2001，40(1)
8. 徐皑冬. 基于以太网的工业控制网络[J]，信息与控制，2000，29(2)
9. 周晓兵. 以太网在工业自动化领域的应用现状和发展前景[J]，自动化仪表，2001，22(10)
10. M.P.de Albuquerque，E.Lelievre-Berna. Remote monitoring over the Internet[J]，Nuclear Instruments and Mwthods in Physics Research A，1998，412:140-145
11. Jonathan D' Cunha，MD，PhD，Cathryn E Larson，Michael A Maddaus，MD，FACS，George H Landis，MD. An Internet-Based Evauation System for a SurgicalResidency Program， InternetEvaluation for a ResidencyProgram[J]，ComputerNetworks，2003，196(6):905-910
12. C.L.Churms，V.M.Prozesky，K.A.Springhorn. The remote control of nuclear microprobes over the Internet[J]，Beam Interactions with Materials&Atoms，Vol. 196，1999:124-128
13. 鲁士文. 计算机网络原理与网络技术[M]，北京机械工业出版社，1996，179-293
14. 郭胜江，陈朝阳. 一种基于 DSP 的可插拔式以太网接口设计与实现[J]，电子工程师，2004，30（3）：41－44
15. REALTEKSemicondutorcorpRTL8019ASdatasheet[EB/OL]，http://www. 21ic. com. cn，1996
16. 张胜古. 单片机与 ISA 总线以太网解决方案[EB/OL]，http://www. laogu. com，2001
17. 彭可. 面向控制应用的 TCP/IP 协议的实现[J]，计算机工程，2003，29(6)：167-169.
18. Silicon Laboratories. C8051F021 datasheet [DB/OL]，http://www. xhl. com. cn，2000

19. 新华龙电子. 以太网测控资料［M/CD］，新华龙电子有限公司，2004-07
20. 周晓峰. 单片机上简单 TCP/IP 协议的实现[J]，微电子学与计算机，2004，21(2)：99-100
21. [美]W.Richard Stevens. TCP/IP 详解，卷 1：协议[M]，北京：机械工业出版社，2000
22. Gary R. Wright W. Richard Stevens. TCP/IP 详解卷 2：实现[M]，北京：机械工业出版社，2000
23. 潘仕彬，何铮. 以太网与控制设备的并口联接[J]，计算机测量与控制，2002，10:682-684
24. 关丽荣. 单片机嵌入 TCP/IP 的研究和实现〔D〕，沈阳：沈阳工业大学，2004
25. 陈华. 基于 TCP/IP 网络技术的嵌入式技术研究〔D〕，杭州：浙江大学，2001
26. 李冬冬. 嵌入式智能仪表与 Internet〔D〕，北京：北京化工大学，2004
27. 杨全胜，吴强，王晓蔚，朱怡健. 可接入 Internet 的智能仪表的设计[J]，工业控制计算机，2001，14(12)：4-5
28. 栗大超，宋光德，靳世久. 嵌入式系统的 Internet 互连技术[J]，微机算计信息，2000，169(6)：4-5
29. 黄松，曾田. 嵌入式网络计算机和 Web-chip 技术[J]，计算机与数字工程，2003，31(1)：33-36
30. 周立功等. 单片机数据通讯应用技术[J]，广州周立功单片机发展有限公司，2003
31. Comer，D.Computer Networks and Internets[M]，Prentice-Nall，1997
32. J.M.Fuertesa，J.Herreraa，J.P.Arboleda. Communication system for a distributed intelligent controller[J]，Computers in Industry，2000，6：35-42
33. Hand. K，Tom L. Real-Time Systems Need Predictability[J]，Computer Design RISC Supplement，Computers in Industry，2002，3：57-62
34. J. M. Fuertesa，J. Herreraa，J. P. Arboledaa. Communication system for adistributed intelligent controller[J]，Computers in Industry，2000，6：35-42
35. 王道平，张义忠. 故障智能诊断系统的理论与方法[M]，冶金工业出版社，2001
36. 虞和济. 故障诊断的基本原理[M]，北京：冶金工业出版社，1998
37. Kenneth S Lee，Magda El Zarki. Scheduling Real-Time Traffic in IP-Based Cellular Networks[J]，IEEE Network，2002，13(4)：39-47
38. 张靖. 检测技术与系统测试[M]，北京：中国电力出版社 2000
39. 王钧，李红玲. C++Builder 经典范例 50 讲[M]，北京：科学出版社 2002

40. 王士元. C 高级实用程序设计[M]，北京：清华大学出版社，1997

致 谢

首先感谢辛勤培养我的导师王玉涛副教授，是导师的悉心指导和严格要求才使我能够顺利地完成本论文的研究工作。导师渊博的学识、宽宏的胸怀、严谨的治学态度和兢兢业业的敬业精神，使我终身受益。几年来，导师不仅在学术上给我谆谆教诲，而且在生活上也给予我无微不至的关怀。在此谨向恩师致以由衷的感谢和崇高的敬意。

在学习期间，杨刚老师、吴恩庚高级工程师、陆增喜工程师、律德才博士生、郭志衡博士生、李小兵硕士生给予作者以很大的帮助，在此深表感谢。同时也感谢高德才硕士生、严其艳硕士生、李俊硕士生等的热情帮助和鼓励，和他（她）们在一起工作和学习使我感到非常愉快，终身难忘。

一份特别的感谢献给我慈爱的父母，他们的鼓励与支持是作者能够顺利完成论文工作的重要保证。

再次向所有关心和帮助过我的老师、同学、朋友和我的家人致以最诚挚的谢意和深深的敬意！感谢各位专家、学者对本文的评阅与赐教！

邢诚

2006 年 2 月于东大园

附 录

附录图 A.1 主控制器原理图