

摘要

21世纪是一个以网络为核心的信息时代。随着嵌入式Internet技术的飞速发展,信息技术不断渗透到各行各业,网络和各种信息家电设备已越来越多地出现在人们的生活之中,Echelon公司提出的“Bring the Internet to life”,即“把互联网带入生活”,反映了实现家电上网已成为网络在人们生活领域发展的一种趋势。而如何建立一个高效率、低成本、基于嵌入式Internet技术的家电控制系统已成为当今世界的一个热点问题。

本文以8位微处理器为核心,描述了基于嵌入式Internet的家电控制系统设计方案及其实现方法。论文首先概述了嵌入式Internet技术的发展及应用前景,在比较了国内外几种主要解决方案的前提下,通过对标准TCP/IP的详细分析,结合家电控制系统的特点,提出了一种基于8位微控制器的嵌入式Internet解决方案,并对在8位微处理器上实现TCP/IP协议的技术作了分析研究。随后详细描述了整个系统的整体结构,阐明了嵌入式网络接口模块的硬件构成、以太网接口芯片的驱动方式,着重分析了嵌入式TCP/IP协议栈的构成、协议的内容、实现这些协议的程序模块等,论文还给出了协议的工作流程和相应的软件设计思想。

在完成了嵌入式网络接口模块的基础上,又详细阐述了空调控制器和电风扇控制器的软硬件设计。家电控制器不仅可以完成电器常规控制功能,还能够于嵌入式网络接口模块进行通信,接收上位机发送过来的命令,处理并返回相应参数,实现用户对家电的远程监控。最后以家电控制系统为研究对象,检验该网络平台的运行效果。实际运行结果表明该方案实用可行。

基于本论文方案实现的网络平台,具有成本低、开发简单快速、灵活等特点,对家电远程控制具有一定的意义和参考价值。

关键词: 嵌入式Internet 以太网 远程监控 家电控制 TCP/IP RTL8019AS

Abstract

As known, the 21st century is an information epoch whose core is the network technologies. With the rapid development of the Embedded Internet technology, information technology has been used in almost every field, more and more network and household information appliances are appearing in everyday life. The slogan "Bring the Internet to life" brought forward by the corporation Echelon indicates that it is a trend to realize household information appliances with computer network developments in the household living field. How to develop an efficient, low-cost smart home based on embedded Internet technique is a hot researching topic in the world.

This paper describes the plan of design of the appliances control system based on the embedded Internet and how it realizes concretely on 8-bit microprocessor. This paper sums up the development and the prospect of embedded Internet technology at first. After the introduction of the main solution of several kinds of the embedded Internet technology domestic and international, through the detailed analysis to standard TCP/IP, and combining the characteristic of the appliances control system, this paper puts forward the solution of embedded Internet platform based on the 8-bit microprocessor, and analyzes the technology of embedded TCP/IP protocol. Then this paper describes the whole structure of the system in detail, illustrates the hardware construction of the interface of embedded network module and the driving method of Ethernet interface chip, and analyzes the composition of embedded TCP/IP protocol, the content, each procedure module of realizing these protocols, etc., and furnishes with the relevant flow chart of procedure.

On the basis of the interface of embedded network module, this paper expounds the design of air-condition controller and electric controller. The controller of appliances not only can accomplish the general control,

but also can communication with the interface of embedded network module, receives the control messages to dispose and return the correctly parameters, realizes remote monitoring and control of the appliance. Finally, the system of appliances control is as the research object to examine the result of this network platform. Through the actual operation result of the system, it proved the feasibility and practicability of this scheme.

The embedded Internet system based on the 51 MCU has the characteristics of low-cost, simplicity, and high-adaptability in the applications, and has a good reference value to the build of remote monitoring and control of the appliance.

Key Words: Embedded Internet, Ethernet, Remote monitoring and control, Appliance control, TCP/IP, RTL8019AS

郑重声明

本人的学位论文是在导师指导下独立撰写并完成的，学位论文没有剽窃、抄袭等违反学术道德、学术规范的侵权行为，否则，本人愿意承担由此产生的一切法律责任和法律后果，特此郑重声明。

学位论文作者（签名）：陈凤帆

2006年5月30日

第 1 章 绪论

当前我们每个人都生活在一个满是嵌入式的世界里^[1]，不管是在科研设备中还是在微波炉中，都可以看到嵌入式系统的影子，嵌入式系统已经成功的应用在各种领域中，并且越来越广泛的进入到人们的生活中。

嵌入式系统发展至今已经有 30 多年的历史，小到一个芯片，大到一个标准的 PC 板，种类繁多，让人目不暇接。嵌入式技术在经历了以单芯片为核心的可编程控制器形式的系统和以嵌入式微处理器为基础、以简单操作系统为核心的嵌入式系统以及以嵌入式操作系统为标志的嵌入式系统这三个阶段后，而今正在进入以 Internet 为标志的嵌入式系统^[2]。这是一个正在迅速发展的阶段，目前大多数嵌入式系统还孤立与 Internet 之外，但随着 Internet 的发展以及 Internet 技术与家电控制、工业控制技术结合日益密切，嵌入式设备与 Internet 的结合将代表嵌入式系统的未来^[3]。

毫无疑问，未来将是嵌入式 Internet 的时代，各种家用设备实现 Internet 的接入是未来家庭网络的发展趋势。家电上网是家电在功能上的重大扩展，并提高了家电的性能，更加方便地满足了人们对家庭自动化和信息化的要求，有效地提高了家庭生活质量。主要表现在两个方面：（1）可实现家用电器的远程控制。用户通过网络远距离控制自己家中的电器；（2）可实现家用电器的远距离数据采集、诊断和监控。随着嵌入式 Internet 技术的发展，传统 PC 的 WEB 服务器完全能通过嵌入式控制器实现，正如早已为人熟知的口号“Bringing the Internet to life”，即“把互联网带入生活中”，使用嵌入式 Internet 技术实现家电上网是目前网络发展的一个重要方向和必然结果。

1.1 嵌入式 Internet 的发展及其关键问题

嵌入式 Internet 是近几年发展起来的一项新兴概念和技术，是指电子设备通过嵌入式模块而非 PC 系统直接接入 Internet，并以 Internet 为介质实现信息交互的技术，通常又称作非 PC 式 Internet 接入。

Internet 带来了一个新时代，回顾一下 Internet 的发展和应用历史，预测 Internet 的明天对 IT 企业的战略设计意义重大（吕京建，2001）。Internet 的发

展与应用经历了并且正在经历着如下几个阶段^[8]：

(1) 1980年~1990年，Internet发展的萌芽阶段。

在这一阶段，实现了异种计算机之间的联网和信息交换，采用TCP/IP协议进行各种信息的交换，主要解决专业领域如银行、军用系统等采用不同操作系统的大、中、小型计算机的联网问题，这时并不称之为Internet，TCP/IP网络仅是少数计算机专家的概念，这是Internet发展的萌芽阶段。

(2) 1990年~2000年，PC机作为客户机，Internet席卷全球。

在这一阶段，在TCP/IP协议网络上发明的email和www普遍应用，Internet国际互联网被大众接受，这时一个重要的条件是PC普及得非常广泛，即形成了Client/Server(客户机/服务器)体系结构，进而发展为Browser/Server(浏览器/服务器)结构，这时的客户机是已经广泛普及的PC，而服务器是相对复杂的、价格昂贵的超级计算机，即我们所谓的“胖服务器”。在这一时期，Internet就象一场革命改变了世界。

(3) 2000年~2010年，嵌入式Internet时代。

在这一阶段，人们将给地球披上一层“电子皮肤”，嵌入式片上系统成为瘦服务器。预测未来Internet将向何处去，这是全世界科学家关心的问题，包括美国贝尔实验室总裁Arun Netravali的一批科学家对此做出了预测：在这一阶段“将会产生比PC时代多成百上千倍的瘦服务器和超级嵌入式瘦服务器，这些瘦服务器与我们这个世界你能想到的各种物理信息、生物信息相联接，通过Internet自动地、实时地、方便地、简单地提供给需要这些信息的对象”。

嵌入式Internet的实现必须解决两个关键问题，一是实现嵌入式系统接入Internet的技术问题，二是嵌入式系统接入Internet的成本问题^[7]。

嵌入式系统接入Internet同PC机接入Internet一样，必须通过相应的通信协议。目前的Internet采用TCP/IP协议，因此嵌入式系统接入Internet最终必须通过TCP/IP接入，嵌入式系统对信息进行TCP/IP协议处理，使其变成可以在Internet上传输的IP数据包。

Internet的通信协议对计算机的存储器、运算速度等要求比较高，传统的TCP/IP协议通常以PC机为基础来实现，这些计算机具有强大的处理能力和足够的存储空间，而且有许多操作系统可以运行，可以方便的实现网络协议。而在嵌入式

Internet 中,实现网络协议通常由嵌入式处理器来实现。这些处理器除了部分功能较强的 32 位处理器以外,大量存在是功能相对简单、价格低廉的 8 位微控制器。它们的处理速度通常较低,可用 ROM、RAM 和总线宽度都有限,支持 TCP/IP 等 Internet 协议将占用大量系统资源,因此这些系统是很难支持 TCP/IP 等 Internet 协议的。而且各个厂家的嵌入式处理器的指令系统也各不相同,不能像 PC 机那样通过标准的软件、硬件接口接入 Internet 网络。可见,嵌入式系统接入 Internet 的技术关键在于如何结合微处理器本身的特点对通信协议进行编程和设置相应的物理层接口。

另外一个关键是嵌入式系统接入 Internet 的成本问题,如何低成本地实现嵌入式 Internet 直接关系到嵌入式 Internet 应用的广泛程度。嵌入式 Internet 可能应用于昂贵的大型工业设备上,更有可能应用于大量存在的价格低廉的家用电器上。对于普通的家用电器而言,低成本的解决方案尤为重要。《数字化生存》的作者尼古拉·庞帝来中国讲演时,曾预言未来 PC 的市场份额将减少,可能会出现许多非常便宜的因特网接入设备,他建议的价格只有一美元。因此,研究低成本下实现嵌入式 Internet 技术将是嵌入式 Internet 发展的另一个关键。

1.2 嵌入式 Internet 的研究背景、意义及应用前景

众所周知,近二十年来是 PC 机领导业界潮流的时期,从 PC 机到局域网、再到全球互联网,PC 业发展已趋于成熟,现正步入将 PC 机成熟技术向嵌入式产品转化的后 PC 时期。目前,嵌入式系统已广泛渗透到人们的工作生活中。从家用电器、信息终端、手持通信设备到航空航天、仪器仪表、汽车、制造工业、过程控制等领域,嵌入式设备已随处可见^[1]。根据美国嵌入式系统专业杂志 RTC 报道,21 世纪初的十年中,全球嵌入式系统需求量具有比 PC 市场大 10 至 100 倍的商机。1998 年在芝加哥举办的嵌入式系统会议上,与会专家一致认为:21 世纪嵌入式系统将无所不在,它将为人类生产带来革命性的发展,实现“PCs Everywhere”的生活梦想。

另一方面,近几年来 Internet 已经成为人们获得各类信息、相互交流的重要渠道,也使人类实现异地信息共享的愿望成为了现实。网络的普及和成熟已经让人与人、人与电子设备、电子设备与电子设备之间的交流变得更加紧密更加轻松,不再受距离的限制。目前大多数嵌入式系统还处于单独应用的阶段,以微控制器为核心,与一些监测、伺服、指示设备配合实现一定的功能。在一些工业和汽车应用中,为

了实现多个微控制器之间的信息交流,利用 CAN, RS-232, RS-485 等总线将微控制器组网,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。如何实现无所不在的嵌入式设备与 Internet 的结合,使信息的存取与获取更快捷、方便则又成为了人类的第二个生活梦想。

国际上,嵌入式系统接入 Internet 始于 1994 年,初期的进展比较缓慢。近几年来,由于微电子技术的迅猛发展,嵌入式处理器的性能大大提高,价格也逐渐降低;与此同时,Internet 迅速普及,网络通信的性能、质量和可靠性得到稳步提高。目前不少国家的科研机构 and 各大公司看到嵌入式设备接入 Internet 的巨大商业价值和 market 潜力,纷纷投入力量进行研究,从而使得嵌入式 Internet 的研究和应用呈现快速发展的势头。

我国正处在 Internet 的初级阶段,Internet 正在以前所未有的速度向前发展,如果初级阶段称作 Personal to Personal 时代,那么未来的时代将是 Personal to Device, Device to Device 时代,随着该时代的到来,智能将下移到设备,大量的嵌入式设备将连到 Internet 上,并通过网络相互通信,自动操作,而无需人的干预,可以想象,随着仪表、控制器、家电等嵌入式设备连到 Internet 上,网上设备将是现在的数百倍,数千倍。与此同时,随着制造工艺的进步,智能控制芯片和智能传感器的成本越来越低,功能越来越强大,这使微处理器与 Internet 联接的费用降低到了完全可以接受的程度,因此,嵌入式 Internet 在不久的将来必将具有十分广阔的应用前景和广阔的市场。网络专家预测,将来在 Internet 上传输的信息中,将有 70% 的信息来自小型嵌入式系统^[6]。未来的 Internet 技术将是嵌入式 Internet 占主导地位,因此嵌入式系统与 Internet 的接入方式已成为人们研究的热点。

从嵌入式 Internet 的应用来看,它的服务对象通常是嵌入式系统,它为嵌入式系统提供 Internet 接入功能。当然也可以为一些没有内含处理器的电器设备提供网络接入服务。总之,信息家电、网络仪器、网络测量、网络监控技术等实现都离不开嵌入式 Internet 技术^{[4][5]}。

- (1) 嵌入式 Internet 解决了计算机以外的终端设备网络化问题
- (2) 嵌入式网络接入技术为终端设备产品提高竞争力创造条件
- (3) 嵌入式 Internet 为工业控制领域设备的发展提供动力
- (4) 嵌入式 Internet 为信息家电和智能家居提供了技术保证

嵌入式系统接入 Internet 技术具有广阔的应用前景，其主要应用领域有^[2]：

(1) 信息家电

21 世纪是数字化网络的时代，计算机、通信和控制 3C 技术相互融合，各类消费类电子产品随着数字化技术与互联网的普及而走向数字化、网络化。嵌入式计算机是实现电子产品数字化、网络化的技术基础。随着数字化进程的日益加深，人们的家居生活越来越明显地印上了“网络的烙印”，最突出地表现就是家电开始走向数字化、网络化。

(2) 智能大厦

智能大厦是信息技术与房地产相结合的产物。随着计算机的普及应用以及网络和自动化技术的发展，大厦内所有公共设备都将采用嵌入式智能系统来提高大厦的服务能力，嵌入式系统的应用可实现大厦内各种操作和信息的共享，实现按需控制。

(3) 智能仪器

参考 PC 机主板设计方法设计的嵌入式计算机智能仪器应用系统通用性强，并可根据不同要求，选择其中的全部或部分电路，只需对软件做部分修改或删除，简化系统，大大提高了开发效率，降低了开发成本，具有广阔的应用空间。

(4) 工业自动化

工业现场应用了大量的 8、16、32 位嵌入式微控制器，其网络化是提高生产效率和产品质量、减少人力资源的主要途径，在工业过程控制、电力系统、电网安全、电网设备监测、石油化工等系统中具有广阔的应用前景。

(5) 虚拟现实中的应用

如虚拟家政系统，水、电、煤气表的自动抄表，安全防火、防盗系统；虚拟智能公路，交通管理、车辆导航、流量控制、信息检测与汽车服务；虚拟精品店、虚拟超市、虚拟库房等等，客户可以在 Internet 上实时地看到存货状况。

1.3 本文研究内容

普通的 8 位微控制器为核心的嵌入式系统随处可见，它应用的数量远远超过高档的嵌入式处理器。在各类嵌入式系统中，8 位微控制器所占的比重最大，因此，在 8 位微控制器上实现联网功能尤为重要。摩托罗拉微控制部消费和工业品市场拓展部经理 Renee Mitchell 曾指出：“日常用品连接因特网的需要是保持 8 位 MCU 市场活力的动力之一，这一联网需求将成为未来几年里支撑 8 位 MCU 市场持续增长的

主要推动因素。”但是这些8位处理器的处理速度较低，而且其内部的随机存储空间也一般只有几百个字节。这使得在这种处理器上实现TCP/IP协议将面临很大的挑战。

通过对嵌入式Internet相关技术的研究，同时结合家电控制系统的具体情况，本文设计了一个基于8位微控制器的嵌入式Internet模块，该模块能够通过串行口与家电控制器通信，实现用户对家电的远程控制、管理和升级等功能，改变家用电器以往单独、孤立的存在方式，进入一种开放、互连的方式。本文所研究的主要内容安排如下：

第1章 绪论。首先介绍了嵌入式Internet技术的基础概念，然后讲述了嵌入式Internet的研究背景、意义及应用前景，最后简述了本论文的主要研究内容。

第2章 系统总体方案设计。分析了国内外几种主要的嵌入式Internet技术的实现方案，通过比较各种方案的优缺点，提出本论文的实现方案，然后分析了其实现的可行性。

第3章 嵌入式网络接口模块硬件设计。详细描述了嵌入式Internet技术的实现方法，实现了嵌入式网络接口模块的硬件设计，为家电控制系统的实现奠定了硬件基础。

第4章 嵌入式网络接口模块软件设计。详细阐述了家电控制系统的软件实现，主要论述了嵌入式TCP/IP的实现，在不影响网络正常通信功能的前提下对标准TCP/IP协议进行了合理的简化。

第5章 家电控制器的设计与实现。以电风扇和空调远程监控系统为应用对象，详细描述了网络电风扇和网络空调的软硬件设计，最后对家电控制系统进行测试，并给出了运行结果。

第6章 结束语。对论文进行总结，提出了一些尚待研究的问题，并对下一步的工作进行了展望。

第2章 系统总体方案设计

本章首先对网络接入问题进行讨论,然后介绍了目前国内外嵌入式 Internet 的实现的几种主要方案,并对它们各自的优缺点进行了比较,接着针对普通 8 位微控制器接入 Internet 的问题进行了探讨,具体从网络接口和网络协议实现问题两个方面分析了其实现的可能性。

2.1 Internet 接入问题

接入Internet有多种方式,如电话拨号接入、ISDN、ADSL、以太网等。其中以以太网作为一种廉价高效的Internet接入方式,已经得到了非常广泛的应用^[10]。

目前,以太网已成为局域网公认的标准。比如在目前的智能小区,每个家庭不能布很多线,而以太网双绞线是一定有的。宽带上网是直接通过以太网的,如果设计的设备能够走以太网的话,可以利用现成的以太网网络。但如果走其它网络,如 RS485, CAN, LONWORKS等,那么需要另外布线。从成本看,用以太网实现联网要比 CAN, LONWORKS更为便宜,集线器、交换机现在都非常便宜,而且将来有三网(电话、电视和计算机)合一的趋势。将来可能是高速以太网的天下,电话信号、电视信号、联网都在以太网上跑。尽管目前还未能实现,但是这种趋势是不可避免的。

以太网在办公自动化领域和企业管理网络内,由于技术成熟、性能稳定、可靠性高而得到了广泛应用,占据国内现有局域网的80%;在测控领域,它以速度快,协议通用,可扩展性好,易于和Internet集成等特点而被视为继DCS(集散控制系统)、FCS(基于现场总线的控制系统)之后控制网络中最热门的研究领域之一。

与其它接入方式相比,以太网有如下一些优点:

(1) 可靠性高。它使用的是一种很简单又很可靠的传输机制,这种机制每天在世界范围的各个站点之间可靠地传输数据。

(2) 通信速度快,从最初的10Mbps已经发展到现在的10Gbps。

(3) 近期以太网技术的发展使其可以作为工业控制网络已开始得到应用,它可以满足控制系统各个层次的要求,使企业信息网络与控制网络得以统一。

(4) 技术成熟,受支持度最好,可直接利用大量现有产品组网。

(5) 互联性、开放性好,直接与PC等设备相连,节点数量基本上无限制。

(6) 设备成本低。因为有了广泛的应用和相关标准的制定,使得以太网产品

的成本得到大幅度的下降。

(7) 用户拥有成本下降。由于以太网已被使用多年，因此具有大量的软件资源，人们对以太网设计、应用有很多经验，对其技术十分熟悉。

(8) 可支持的传输介质多，有光纤、双绞线、同轴电缆等。

(9) 扩展性好、发展潜力大，以太网的传输速度有10Mbps、100Mbps和1Gbps，甚至10Gbps。对大部分应用来说，10Mbps已算是足够快了，当10Mbps有必要提高至更高速时，可以方便地进行升级，扩展非常容易且可以选择多个制造商的产品。

2.2 嵌入式 Internet 接入方式

利用 MCU 实现嵌入式 Internet 方案的技术难点在于：如何利用 MCU 自身有限的资源对信息进行 TCP/IP 协议处理，使之变成可以在 Internet 上传输的 IP 数据包。从解决这一技术问题出发，目前主要有以下几种方案。

(1) 32 位 MCU + RTOS

采用 32 位高档单片机，在 RTOS（实时多任务操作系统）平台上进行软件开发，在嵌入式系统中实现 TCP/IP 协议处理。目前较为流行的 RTOS 有 VxWorks、pSOS、Nucleus、QNX、Windows CE、Linux 等。此方案的优点：由于采用高档处理器，系统功能强大，能完成很多复杂的功能。缺点是：a. 嵌入式操作系统的运行需要占有大量的处理器资源和存储器空间，对硬件配置有较高的要求，开发成本较高；b. 如果不是采用开源操作系统，需要购买昂贵的嵌入式 RTOS 开发软件，对开发人员的开发能力要求较高。c. RTOS 内部实现的 TCP/IP 协议栈是通用的，针对性不强，难免会有相对于具体应用多余的代码，不能保证代码的尽量精简。

(2) 8/16 位 MCU + 精简 TCP/IP 协议栈

对于大量存在的 8 位和 16 位 MCU，由于其速度较慢，内存较小，如要支持 TCP/IP 协议将占用大量系统资源。考虑到系统功能较为单一，数据量较小，可以根据实际要求将 TCP/IP 协议栈做大幅度的简化，只保留其中核心的部分，这样，就可以大幅度减少对于系统资源的需求，从而可以在低成本、低速度、小内存的 MCU 上实现网络连接，这样的方案的优点是：采用通用器件，成本较低，开发灵活，便于广泛应用；缺点是：a. 开发周期长，对开发人员有较高要求；b. 协议栈需要长期深入测试、修改才能达到可靠性要求。

(3) MCU + TCP/IP 协议芯片

由 MCU 及固化了 TCP/IP 协议的硬件芯片组成应用系统的核心, 应用系统可以直接拨号上网, 硬件电路相对简单。但需要大容量的存储器, 如果所用的 TCP/IP 协议芯片是软件固化的, 还要求 MCU 有较高的运行速度。采用这种方案的芯片有 Scenix Semiconductor 公司的 SX-Stack, Seiko Instrument 公司的 S7600A, iReady 公司的 Internet Tunner, Connect One 公司的 iChip 等, 其中 Scenix 公司的单片机为软件固化协议, 其它芯片为硬件固化。此方案具有开发时间短、系统成本低的优点。但需要熟悉 TCP / IP 等协议和相关接口, 软件设计量较大; 另外每个电子设备都需要一个 IP 地址, 因此需要扩充 IP 协议才能到更好的支持。

(4) PC Gateway + 专用网

采用专用网络(如 RS232, RS485, CAN Bus 等)把一小批单片机连接在一起, 然后再将该专用网络连接到一个 PC 上, 该 PC 作为网关将专用网络上的信息转换为 TCP/IP 协议数据包, 然后发到网上实现信息共享。这种方式可以降低对嵌入式系统的要求, 使用范围广泛; 可以连接多种单片机, 充分利用已经大量存在的 8/16 位嵌入式产品, 集中分散的小型设备成为网络, 并提供使用 Internet 进行信息交互的手段, 有利于设备远程访问和实时监控。但是这种方案存在如下缺点:a. 需要依赖 PC 机作网关进行协议转换, 在多个单片机系统分散的情况下, 专用网络布线极为不便; b. 需要在 PC 机上安装专门的协议转换软件, 这种软件通常由专门的第三方软件商提供, 费用较高。

(5) MCU + EMIT + emGateway 协议

采用 emWare 公司开发的嵌入式微 Internet 网络技术 EMIT(Embedded Micro Internetworking Technology)。EMIT 并不能让设备具备 Internet 的连接能力, 而是在应用系统的 MCU 内部使用 emNet 协议, 再通过网络接口软件 emGateway 网关与 Internet 连接。emGateway 网络接口软件可以安装在计算机、TV 机顶盒或专用的家用电器服务器中, 它支持 TCP/IP 协议并运行 HTTP 服务程序, 用户可以通过网络浏览器访问远程服务器。emGateway 通过 RS-232, RS-485, CAN 红外及射频等通信方式与多个嵌入式设备相连。每个嵌入式设备的应用程序包含一个独立的通信任务, 称为 emMicro, 监测嵌入式设备中预先定义各个变量, 并将结果反馈到 emGateway 中, 同时 emMicro 还可以解释 emGateway 的命令, 修改设备中的变量, 或进行某种控制。为了完成网络连接, emWare 公司开发了大量软件来建立访问和监控 MCU 应用

系统的接口，这些接口可以在不同的设备中。emMicro 和 emGateway 一起为嵌入式设备提供了 Internet 功能。这种方案具有一定的性价比优势，MCU 只使用较简单的 emNet 协议，对 MCU 的要求较低。但应用系统设计工程师必须熟悉 emNet 协议和相关的接口，原来客户应用系统的 MCU 也不一定符合要求，并且硬件设计的工作量仍然较大。同时，应用系统的单片机处理 emNet 协议要占用一定的系统资源。

对于家电远程监控系统，数据传输量一般不大，而且对实时性要求不高，其时延可以在 1 秒以上。使用性能较高、功能齐全的 8 位微处理器、再加上功能强大的协议芯片完全可以满足应用需要。因此，本设计选用方案 2 进行系统设计，即采用 8 位微处理器和网络接口芯片相结合的方法，在微处理器上实现 TCP/IP 协议栈及以太网控制芯片驱动程序。采用这种设计方案进行系统设计，既简洁，又经济。

2.3 网络接口控制器数据传输分析

2.3.1 网络接口问题

以 10BaseT 以太网为例，发送数据时应该做的工作是，首先对需要发送的数据进行曼切斯特编码，然后对编码后的数据进行预扭曲处理。使其发送的数据适合在 10BaseT 的以太网络上传输，最后把处理好的数据以 10M 的速度发送到以太网络上。同时，为了保证数据的有效发送，系统还应具有冲突检测和重发的功能。

从以上的数据发送过程可以看出，直接用普通的微处理器来实现上述功能非常困难的。解决这个问题的方法是采用专门的网络接口芯片 NIC 来实现，这类芯片要遵循 IEEE802.3 所规定的 CSMA/CD 协议，除了提供物理链路所需要的电气性能外，还提供曼切斯特编码、冲突检测和重发的功能，它可以与很少的外围电路一起完成数据的发送和接收功能。处理器只需为 NIC 芯片提供初始配置和数据接口即可。

2.3.2 网络通信协议问题

解决网络接口问题之后，还要考虑的是如何实现众多的网络通信协议，我们可以把不同协议层的数据统一理解为长短不一的二进制数据流。数据的发送可以看成按照一定格式生成不同长度的数据流。数据的接收可以看成按照一定的规则从不同长度的数据流中得到需要的数据^[7]。由此看来，处理器实现 TCP/IP 协议就是实现数据流的生成及解析工作，下文将从数据发送和数据接收两方面来论述采用 8 位微处理器来实现 TCP/IP 协议的可能性。

(1) 数据发送的封装问题

把一组数据发送到基于TCP/IP协议的网络上，首先是产生符合TCP/IP协议要求的数据格式，然后把它按照一定的规定发送到网络上去。首先看一下如何产生符合TCP / IP协议的数据格式。

以太网接入Internet，首先从一个物理帧的格式来分析。一个标准的IEEE802.3的物理帧格式如表2-1所示：

表2-1 IEEE 802.3的物理帧格式

帧头	帧起始定界符	目的物理地址	源端物理地址	数据域长度	数据域	校验和

前文已经提到，可以使用专门的NIC芯片来实现以太网链路层的数据收发功能，NIC芯片除了能够实现以上功能外，还可以实现在发送的物理帧上自动添加帧头（Preamble）、帧起始定界符（SFM）和校验和（checksum）的功能。因此处理器只要能够产生“目的物理地址+源端物理地址+数据域长度+数据域”的数据流即可。

处理器可以采用“填充法”来减少形成数据流的时间，“填充法”是把不太改变的数据部分事先准备好（如源端物理地址等）。然后把改变的部分（如数据域）填充进去即可。

不同层的通信协议方式从处理器的角度来看无非是“数据域”的不同而已，可以用同样的方法进行封装。数据的封装对于处理器没有严格的时间限制，普通的8位微处理器完全可以胜任。

数据流的形成过程中还有一个问题需要处理器来解决，就是计算校验和，这是因为很多上层协议的数据封装中都需要有校验和一项，例如IP协议、TCP协议等。校验和的计算采用以下方法：即把整个需要计算校验和的部分看成是由一串16 bit的字组成，对每个16 bit项进行二进制反码求和即可得到校验和。这种反码求和运算在处理器中很容易实现。

(2) 数据接收的分析

数据接收部分的基本任务首先是判别收到的数据包是否为自己的数据报，然后从不同协议封装的数据报中获取需要的信息（包含协议信息和数据信息），以此信息进行下一步的所规定的动作。

判别是否为自己的数据报可以从接收的数据帧的物理地址标识和IP地址标识来判别，这只是一些简单的逻辑判断。

从不同协议封装的数据报中获取需要的信息首先能够判别数据报的不同协议，其次是熟悉各种协议的封装结构，从而进行不同的处理操作。这两个方面是可以由处理器执行多次判断来完成，判断的难度取决于协议的复杂程度。

从上文的论述可以看出，在8位微处理器上实现TCP/IP协议是完全有可能的。当然，TCP/IP协议族中有上百个协议，真正实现嵌入式Internet还需要多方面的技术，这些技术包括协议的选择、协议的简化、不同协议的具体实现等多方面的工作，本文的后续部分还将就这些问题给出详细的分析。

2.4 系统整体结构

从家电控制系统的远程在线访问的角度考虑，为了实现家电接入 Internet，设计了如图 2-1 所示的家电控制系统。本系统主要是由客户端、嵌入式网络接口模块和家电控制器模块三部分组成。

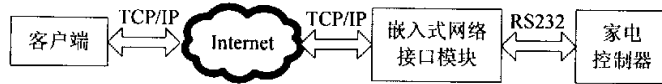


图 2-1 家电控制系统结构图

图中客户端是系统中的控制端，是远程用户和该系统的接口部分，通过它用户可以和系统进行远程交互，从而完成各种相关操作，任何一台连入网络的计算机均可以作为客户端；嵌入式网络接口模块相当于一个控制中心，负责与 Internet 相连，传递客户端和家用电器控制模块之间的控制信息，承担着信息发送和接收的任务；家电控制器模块实际上就是系统中的受控点，是用户要操控的对象，接收来自远程用户的命令，对家电进行相关的控制操作。

2.4.1 客户端

由于嵌入式网络接口模块可以将家电的信息发布到 Internet 上，那么可以在网络的另一端通过浏览器访问家电，获取家电的信息，并对其实施控制。

客户端的用户主要通过可视化的操作界面对家电进行控制，该操作界面是 Web 页面，一般是通过存储在嵌入式网络接口模块中的中间信息分析而形成的。用户在客户端键入某个家电的地址就能够登录到该家电的操作界面，通过点击界面上的按钮向家电发出控制命令或得到家电的运行状态。当用户向家电发出控制命令时，所发出的控制指令由该网页所在的控制中心服务器进行封装，通过 Internet 到达该用户的嵌入式网络接口模块上。嵌入式网络接口模块收到数据后进行拆封，将控制命

令发送给家电控制器，家电控制器读取并执行相应的操作，当选择的操作有返回值的时候，嵌入式网络节点将结果发送到客户端，并在操作界面上显示。

2.4.2 嵌入式网络接口模块

嵌入式网络接口模块是远程客户端通过 Internet 访问家电的中转站，它可以将来自互连网络的各种信息通过协议转换的方式传递到家电控制器，使远程控制点可以对家电进行控制。嵌入式网络接口模块应该具有如下的一些基本功能：

(1) Internet 接入口

作为外部 Internet 和家电控制模块的连接点。客户端通过 Internet 访问嵌入式网络接口模块，来完成对家电的监控；而家电控制模块又可以通过它接入 Internet，实现各种应用。

(2) 协议转换

必须能够实现本地通讯协议与 TCP/IP 数据包之间的映射，完成不同传输介质和网络拓扑结构间信号的转化，实现不同类型网络之间的连接。

(3) 设备控制

一方面，实现对各种家电资源的获取，完成对室内外环境的数据采集，然后根据客户端用户的访问请求发送这些家电的信息；另一方面，当用户发出控制信息后，将接收到的指令提交到家电控制模块执行，并接收从家电控制模块返回数据，刷新页面。

(4) Web 服务器

具有超文本传输协议 HTTP 发送的能力，可以将家电信息转换成操作页面，具有浏览器功能。用户进入电器的控制选项界面，进行控制家电，操作简单。

2.4.3 家电控制模块

家电控制模块除具备常规控制功能外，还应具备与嵌入式网络接口模块的通信功能。所以在其内部包含一个功能实现模块，对家电的每一个功能都给出具体的实现。还包括了一个网络通信模块，按照一定的协议与嵌入式网络接口模块进行通信。

根据系统的特点，及对数据传输的要求，嵌入式网络接口模块与家电控制器之间采用串口进行通信。串口通信有同步传输和异步传输两种方式，本设计中采用的是异步传输方式，主要是因为异步传输方式有如下的一些特点：

(1) 异步传输简单，双方时钟可允许一定误差。同步传输较复杂，双方时钟

的允许误差较小。

(2) 异步传输适合于小量的、短时间的数据传输。同步传输适合于大量的、长时间的数据传输。

2.5 电磁干扰 (EMI) 技术

电磁干扰抑制对家电控制系统的设计至关重要, 因此, 加强电磁兼容 (EMC), 提高 EMI 抑制技术, 成为系统主要考虑。

电磁干扰是指由外部噪声和无用电磁波在接收中所造成的骚扰。一个系统或系统内某一线路受电磁干扰程度可以表示为如下关系式:

$$N=G \times C / I \quad (3-1)$$

G: 噪声源强度;

C: 噪声通过某种途径传到受干扰处的耦合因素;

I: 受干扰电路的敏感程度。

C、G、I 这三者构成电磁干扰三要素。电磁干扰抑制技术就是围绕这三要素所采取的各种措施, 归纳起来就是三条: 抑制电磁干扰源; 切断电磁干扰耦合途径; 降低电磁敏感装置的敏感性。下面就这三方面分别做出分析。

(1) 抑制干扰源

要想抑制干扰源, 首先必须确定何处是干扰源, 在越靠近干扰源的地方采取措施, 抑制效果越好。一般来说, 电流电压剧变即 di/dt 或 du/dt 大的地方就是干扰源; 具体来说继电器开合、电容充电、电机运转、集成电路开关工作等都可以成为干扰源。另外, 市电电源也非理想的 50Hz 正弦波, 而是充满各种频率噪声, 是个不可忽视的干扰源。抑制方法可以采用低噪声电路、瞬态抑制电路、稳压电路等; 器件的选择则尽可能采用低噪声、高频特性好、稳定性高的电子元件。

(2) 切断电磁干扰耦合途径

电磁干扰耦合途径主要为传导和辐射两种。噪声经导线直接耦合到电路中最常见的。抑制传导干扰的主要措施是串接滤波器。根据信号与噪声频率的差别选择不同类型的滤波器。如果噪声频率远高于信号频率, 常采用 LC 低通滤波器, 这种滤波器结构简单, 滤除噪声效果也较好。

对于通过供电电源线传导的噪声可以用电源滤波器来滤除; 对传输线路及印刷电路板的布线设计, 应注意进线与出线、信号线与电源线尽量分开。对于重点线路

可采用损耗线滤波器、三端子电容、磁环等器件进行干扰抑制；对于接口端，可加连接器；对于辐射干扰，主要措施是采用屏蔽技术和分层技术。

(3) 降低电磁敏感装置的敏感度

电磁敏感装置是一柄双刃剑，一方面人们希望接收装置灵敏度高，以提高对信号的接收能力；另一方面，灵敏度高受噪声影响的可能性也就越大。因此，根据具体情况采用降额设计、避设计、网络钝化、功能钝化等方法是解决问题的方法。

综上所述，对于电磁干扰的抑制方法很多，设计时应选择适当的电磁干扰技术。

2.6 系统评价指标

(1) 控制准确性

这是系统的第一要素。系统运行时，只有对设备准确无误地进行控制，系统才是有意义的。在 IPV4 网络环境正常的情况下，系统对家电进行控制时，要求控制准确率 100%。

(2) 时间特性

系统正常运行时，要求系统的响应时间小于 2s。

(3) 适应性

系统应根据用户要求灵活调整监控系统的设备的种类。

2.7 本章小结

本章首先对网络接入问题进行了讨论，然后简要介绍了目前国内外嵌入式 Internet 实现的几种主要方案，并对它们各自的优缺点进行了比较。接着根据家电控制系统的特点，针对 8 位微处理器接入 Internet 的问题进行了探讨，从网络接口和网络协议实现两个方面分析了其实现的可能性。然后阐述了家电控制系统的整体结构及其各个模块的特点和功能。最后论述了设计时应考虑的电磁干扰问题，以及系统的评价指标。通过对本章的研究，为后续章节具体实现家电控制系统打下了理论基础。

第3章 嵌入式网络接口模块硬件设计

嵌入式网络接口模块的功能是存储监控页面并进行数据传输，本系统人机交互界面的源程序即存在于此，客户在任何一台连入网络的PC机网络浏览器中键入预设的IP地址就可调出该系统的主页面，从页面中的相关按钮可链接到家电监控页面，进行相关处理。在传输数据方面，嵌入式网络接口模块将来自于家电控制模块的实时工作状态信息传输到客户端显示，也将用户在控制页面中输入的控制信息传输到家电控制模块中以使家电进行实时动作。本章将详细介绍嵌入式网络接口模块的硬件设计。

3.1 硬件结构图

针对大多数家电远程在线访问的要求，对它们的监控一般只要满足下列两个条件即可：（1）远程访问家电时的通信数据量不大。（2）家电的远程监控对实时性要求不高可延时在1s以上。基于以上两个条件，在设计时，选择了单片机SST89E554RC作为家电接入Internet的微处理器，同时采用芯片RTL8019AS作为以太网的驱动芯片，负责帧的接收和发送。硬件结构图如图3-1所示，可以看出嵌入式网络接口模块主要有MCU模块、以太网控制模块以及串行通信模块组成。

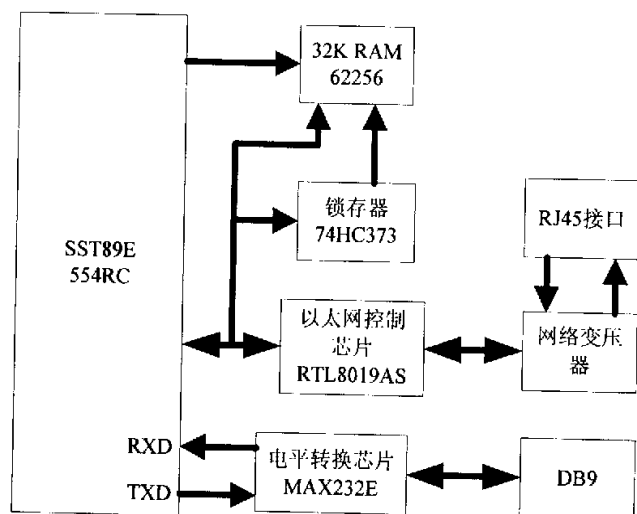


图3-1 嵌入式网络接口模块硬件结构图

3.2 MCU模块

MCU模块是整个系统的核心，主要包括主处理器和RAM扩展电路。这里主处理芯

片选用SST公司的SST89E554RC, 该芯片是一种高性能、低功耗、采用CMOS工艺制造的8位微处理器, 在软件、封装和引脚上和8051完全兼容, 最高时钟频率达40MHz, 且片内硬件资源较为丰富, 拥有40K的Flash存储器和1K的RAM, 如此大的片内存储空间使得系统不再需要片外存储空间, 简化了系统的复杂度。其内部的Flash支持IAP编程, 非常便于现场或远程软件修改、调试或升级, 也可以用于保存数据。

由于嵌入式网络模块要处理以太网数据包, 并且要实现TCP/IP、HTTP等协议和其他应用程序, 所以必须扩展RAM, RAM扩展电路选用了一片74HC373和62256扩展了32K字节的RAM, 占用了单片机外部数据地址空间0000H~7FFFH。使用外部RAM的目的是提高单片机的数据传输速度和进行复杂的TCP/IP处理。由于以太网的包最大可以有1500多字节, 单片机是无法存储这么大的包, 只有放到外部的RAM里。同时这外部RAM也用作串行口的输入输出缓冲, 以使单片机可以高速的吞吐数据。用RTL8019AS上的RAM来代替62256也可以, 但操作起来比较复杂, 也会影响速度。

3.3 以太网控制模块

以太网控制模块主要是在Ethernet/Internet和微处理器之间起到一个桥梁的作用。将微处理器传出的数据打包送到以太网上, 或将以太网上的数据接收进来, 供主处理器处理。其核心是以太网控制芯片RTL8019AS, 为使RTL8019AS能与8位微处理器一起配合工作, 就要对它工作时的模式、状态及相关参数作硬件的设置。

3.3.1 以太网控制器RTL8019AS

以太网控制芯片RTL8019AS实现了以太网媒介访问层(MAC)和物理层(PHY)的功能, 包括MAC数据帧的组装/拆分与收发、地址识别、CRC编码/校验曼彻斯特编解码、接收噪声抑制、输出脉冲成形、超时重传、链路完整性测试、信号极性检测与纠正等。它具有以下一些特点:

- (1) 支持Ethernet II与IEEE802.3(10Base5, 10Base2, 10BaseT)标准。
- (2) 全双工, 收发可同时达到10Mbps的速率, 自动产生CRC校验码、数据帧同步码和帧起始定界符。
- (3) 内置16KB的SRAM, 用于收发缓冲, 降低对主处理器的速度要求。
- (4) 支持8/16位数据总线, 8个中断申请以及16个I/O基地址选择。
- (5) 支持UTP、AUI、BNC自动检测; 支持对10BaseT拓扑结构的自动极性修正。
- (6) 允许4个诊断LED引脚可编程输出。

(7) 100脚的PQFP封装, 减小了PCB尺寸。

3.3.2 RTL8019AS工作模式与网络接口模式设置

(1) 工作模式设置

RTL8019AS有3种工作方式: 第一种为跳线方式, 其I/O基址和中断由跳线决定; 第二种为即插即用方式, 由软件自动配置相应参数; 第三种为免跳线方式, 其I/O和中断由外接的93C46的内容决定。在嵌入式应用场合, 如果不使用93C46, 可以降低成本, 同时又减少连线。故本设计选择第一种工作方式。

RTL8019AS的工作模式由第65脚JP决定, 当该引脚为低电平时, RTL8019AS是工作在第2/3种方式, 由93C46里的设置内容决定。在一般应用场合, 网卡中的RTL8019AS的JP脚悬空, 输入状态为低电平(其他引脚也是, 因片内接有一个100K的下拉电阻), RTL8019AS工作于第2/3中工作方式, 需要使用93C46。而此处把第65脚JP接高电平(VCC), 即使用跳线方式, 则RTL8019AS的I/O和中断就不由93C46的内容来决定, 这时并不需要使用93C46。

(2) 网络接口选择

RTL8019AS使用AUI还是BNC接口的引脚, 由AUI(第64脚)决定。常用的网络接口一般是BNC的, BNC接口方式支持8芯双绞线或同轴电缆, 很少用AUI。该引脚高电平时使用AUI接口, 将该引脚悬空即可设置RTL8019AS使用BNC接口。

而网络介质类型由引脚PLO和PL1(第74和77脚)决定, 参见表3-1:

表3-1 网络介质选择

PL1	PLO	网络介质类型
0	0	自动检测(10BaseT连接检测允许)
0	1	10BaseT(双绞线), 连接检测禁用
1	0	10Base5(同轴粗缆)
1	1	10Base2(同轴电缆)

本设计选用了的UTP(非屏蔽双绞线)作为传输介质, 从表中可看出, 应把PL1和PLO两个输入引脚接低电平, 把这两个引脚也悬空。这样, RTL8019AS就工作在带连接检测的TP/CX接口方式下, 会自动检测接口类型是用同轴电缆还是双绞线。

(3) I/O基址、中断线与数据总线模式设置

当RTL8019AS工作于跳线模式时, RTL8019AS的I/O基址由引脚IOS3~IOS0(第

85, 84, 82, 81管脚) 决定, 本设计在此处设置的基址为300H, 实现方法是将对应的IOS0、IOS1、IOS2、IOS3都接低电平。

芯片的中断由IRQ2~IRQ0(引脚80, 79, 78) 决定, 本设计使用查询方式, 没有用中断, 因此中断引脚全部悬空。

RTL8019AS可兼容8位和16位数据总线操作, 由于所用微处理器是8位数据总线, 因此要设置RTL8019AS使用8位总线模式。RTL8019AS的第96脚IOCS16是16位数据总线的选择脚。上电复位时, 为RTL8019AS的输入脚, 如果输入为低电平, RTL8019AS将选择8位模式, 反之选择16位模式。设计中将IOCS16引脚通过电阻下拉至地。上电复位时, 该引脚输入电平为低, RTL8019AS选择8位数据总线模式。

3.4 RTL8019AS与单片机的硬件连接

3.4.1 接口逻辑框图

用微处理器SST89E554RC来对以太网控制器RTL8019AS进行控制和读写操作, 是通过设计两者之间的接口实现的。接口设计主要包括地址和数据总线、复位与读写线的接口设计。其逻辑框图如图3-2所示。

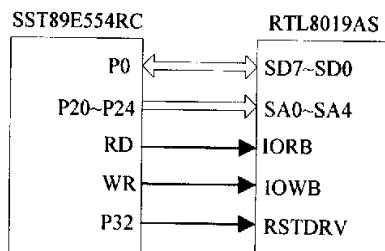


图3-2 SST89E554RC与RTL8019AS接口逻辑图

其中, RTL8019AS的数据线SD0~SD7与SST89E554RC的P0端口相连; 地址线SA0~SA4与处理器的P20~P24相连; 读写线IORB和IOWR连接到处理器相应的读写线上; 第33脚复位引脚RSTDRV与处理的P3.2相连, 程序中用P3.2的输出来复位RTL8019AS, RSTDRV高电平有效, 至少需要800ns的宽度, 一般给该引脚施加一个1us以上的高电平使系统复位。施加一个高电平之后, 然后施加一个低电平。

3.4.2 地址总线接口与地址映射

RTL8019AS具有32位输入输出地址, 地址偏移量为00H~1FH, 根据前述设置, 已经设置RTL8019AS的基址为300H, 则对应于300H~31FH, 300H的地址偏移量为0,

301H的地址偏移量为1, 依此类推。其中00H~0FH共16个地址, 为RTL8019AS的寄存器地址。10H~17H共8个地址, 为RTL8019AS的DMA地址, 18H~1FH共8个地址, 为复位端口, 功能一样, 用于RTL8019AS复位。

RTL8019AS的引脚中, SA0~SA19为RTL8019AS的地址线, 在此使用RTL8019AS的基地址为0300H~031FH, 转换为二进制如表3-2所示。

表3-2 RTL8019AS的寻址地址与地址线关系

引脚	SA19~SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
0300H	0	1	1	0	0	0	0	0	0	0	0
...	0	1	1	0	0	0	X	X	X	X	X
031FH	0	1	1	0	0	0	1	1	1	1	1

从表可以看出, SA19~SA5全部地址线除SA9、SA8接高电平外, 其余全接地, 即是固定的值0000000011000, 只有5根地址线就可以实现对32个I/O端口的寻址, 因此设计中将SA5, SA6, SA7和SA10~SA19都接地, SA8, SA9接高电平VCC, SA0~SA4接处理器的P20~P24, 因此在编写程序时只有SA0~SA4这5根地址线是与处理器的总线交换数据的。

3.4.3 RTL8019AS与传输介质的连接

以太网控制器RTL8019AS与以太网不能直接相连, 要通过网络变压器, 本设计选择了20F001N滤波/变换器, 电路连接如图3-3所示。

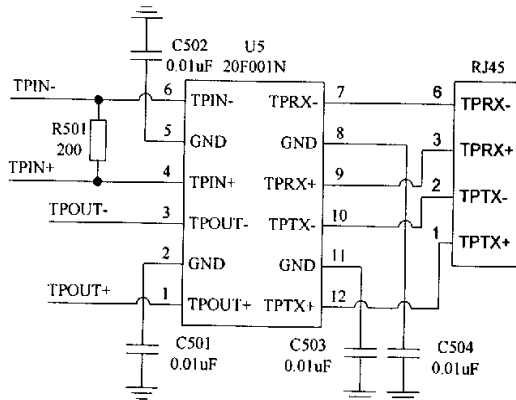


图3-3 双绞线接口电路

20F001N是双绞线驱动/接收器, 为耦合隔离变压器模块, 其内部拥有两个最高频率在17MHz的低通滤波器和两个隔离变换器, 通过它们, 可以去除因为电路的数

字特性导致的高次谐波，这样，UTP电缆在尽可能小的噪声下传输有用信号。

20F001N的引脚1 (TPOUT+)、引脚3 (TPOUT-) 分别接到RTL8019AS的双绞线的差分输出信号引脚45 (TPTx+)、引脚 (TPTx-)。引脚4 (TPIN+)、引脚6 (TPIN-) 分别接RTL8019AS的双绞线的差分输入信号引脚59 (TPRx+)、引脚58 (TPRx-)。

对交流信号，电容C501、C502、C503、C504把变换器的中心引脚连接到地端，以抑制非均衡噪声。

3.5 串行通信模块

串行通信模块是连接家电控制器与嵌入式网络接口模块的桥梁。SST89E554RC带有一个可编程的全双工串行接口，该串行接口具有多级通信功能，可以与PC机、数据收发设备或其他单片机构成通信系统。由于单片机的串行数据接口并不是标准的RS-232-C串口，因此使用了MAXIM公司的MAX232电平转换芯片将单片机的串行数据接口与标准的RS-232-C串行接口连接起来。MAX232是一种能够实现RS232与TTL两种逻辑电平相互转换的专用芯片，芯片内部包含两种接收器和驱动器以及一个电源电压变换器，并且只需要单一的+5V电源供电。MAX232芯片的硬件接口十分简单，单片机的串行接收和发送端RXD、TXD可直接连接到MAX232的相应端口上，通过外接1.0uF的电解电容便可以使MAX232输出RS-232-C串行通信所需要的±10V信号电平。由于MAX232芯片没有片选端，在应用系统中仅起到电平转换的作用，因此它并不占用单片机的外部数据存储空间。其接口电路如图3-4所示。

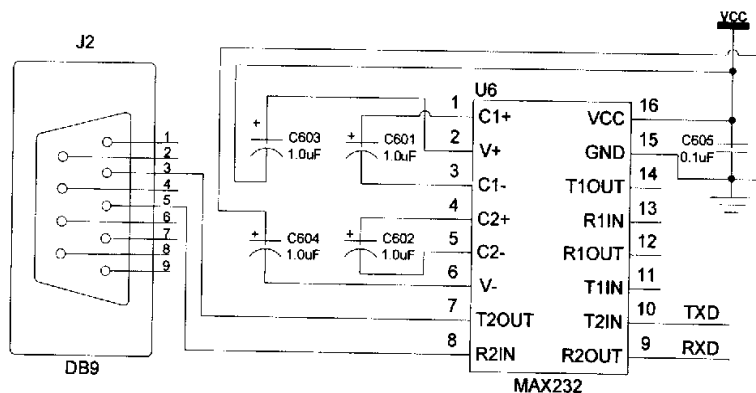


图3-4 串行接口电路

3.6 EMI 设计

电磁兼容性和抗干扰设计，关系到系统的稳定工作，PCB 设计中主要采用以下策略。

(1) 系统中地线结构大致有数字地、模拟地和屏蔽地。各种地线采用“单点接地”的原则，即将电路数字地与模拟地、信号地与电源地均应分开处理，最后接至一个总的系统地。

(2) 电源线、地线尽量粗，信号线尽可能和电源地线并排着走线，这样可以有效地滤除信号纹波。传送信号要与信号回路尽可能靠近，以防止这些线路包围的环路区域产生辐射，并降低环路感应电压的磁化系数。

(3) 电路上每个 CMOS 芯片增加电源去藕电容，减少电源对器件的干扰。对于 PHY 电路使用磁珠，减小大于去藕电容谐振频率的干扰影响。

(4) 晶振振荡电路尽量靠近芯片，用地线把时钟区隔离起来。

(5) RJ45 连接外部网络的地与信号地是两个完全不同的地，应彼此相互隔离。

(6) 对差分信号布线时，差分对的两条线的长度尽可能一样长，两线的间距一直保持一致。

3.7 本章小结

本章首先描述了基于TCP/IP协议的嵌入式网络接口模块的硬件结构，然后对主要模块进行了详细的阐述，主要包括MCU模块、以太网控制模块以及串行通信模块，最后描述了在设计时所采取的抗干扰措施。

第4章 嵌入式网络接口模块软件设计

嵌入式网络接口模块的软件由4个主要部分组成，即系统主程序、RTL8019AS驱动程序、TCP/IP协议栈和协议转换程序。

4.1 系统主程序

系统主程序负责完成系统参数的初始化和实时调用任务。在主程序中采用轮询的方式不断检测状态字，以判断某事件发生与否，如果发生就去处理该事件，事件处理完毕后返回主程序，继续执行后面的程序。

状态字有以下几种：有以太网数据包到达 `ENENT_ETH_ARRIVED`、TCP重传 `EVENT_TCP_RETRANSMIT`、TCP状态保活 `EVENT_TCP_INACTIVITY`、ARP缓冲状态处理 `EVENT_AGE_ARP_CACHE`、ARP重发 `EVENT_ARP_RETRANSMIT` 和 RS232串口缓冲处理 `EVENT_RS232_ARRIVED` 等。

程序开始时，先是系统初始化，主要负责分配内存缓冲池，用于处理TCP/IP协议；接着网络参数初始化；然后依次完成串口初始化、定时器初始化、ARP缓冲初始化、TCP初始化、HTTP初始化和RTL8019AS初始化等工作；最后由一个while死循环检测状态字，有各子程序进行相关处理。主程序处理流程图如图4-1所示。

4.2 RTL8019AS 驱动程序

RTL8019AS驱动程序有两个目的：发送和接收IP报文，发送和接收ARP报文。驱动程序实现的功能就是将要发送的数据包按指定格式写入芯片缓冲区并启动发送命令，RTL8019AS会自动把数据包转换成物理帧格式在物理信道上传输。反之，RTL8019AS收到物理信号后将其还原成数据，按指定格式存放在缓冲区中，以便主机程序取用，简言之就是RTL8019AS完成数据包和电信号之间的相互转换。

RTL8019AS驱动程序有3部分组成：芯片初始化、发送数据包和接收数据包。由4个函数实现：`init_8019()`、`send_frame()`、`query_8019()`和`rcve_frame()`。

4.2.1 RTL8019AS 上电初始化

(1) RTL8019AS 的复位

SST89E554RC的P3.2连接到RTL8019AS的RESDRV脚来进行复位操作。RSTDRV为高电平有效，至少需要800ns的宽度。给该引脚施加一个1 μ s以上的高电平就可

以复位，施加一个高电平后，然后施加一个低电平。复位过程将执行一些操作，至少需要 2ms 的时间，在本系统中延时了 400ms，以确保完全复位。

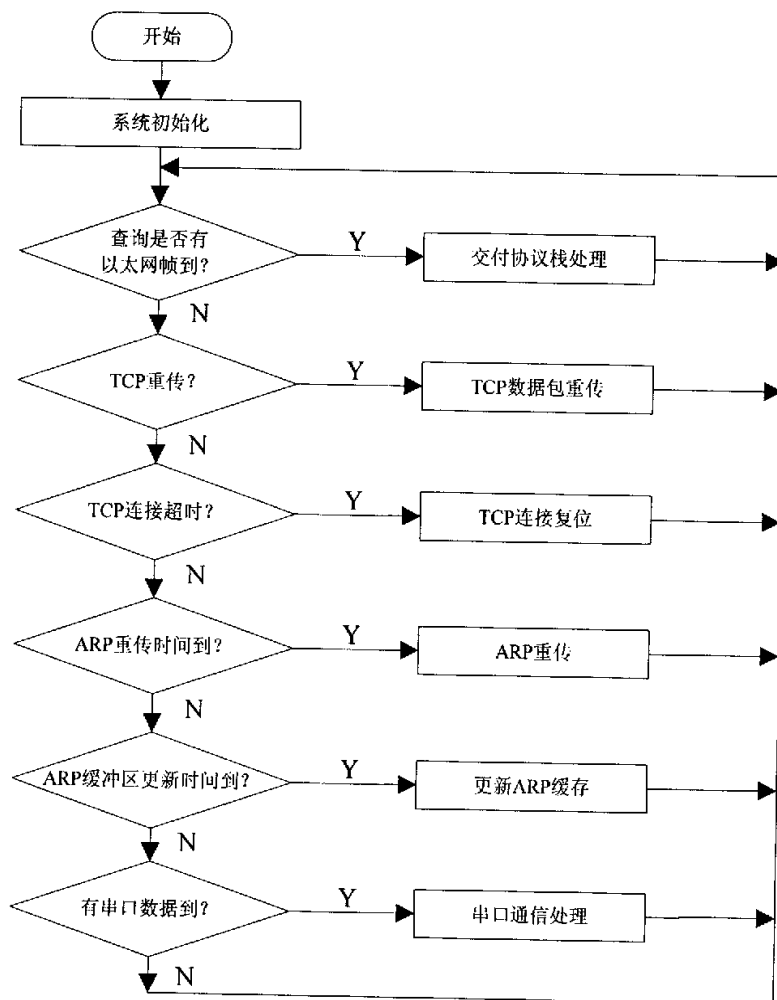


图 4-1 主程序流程图

(2) RTL8019AS 的初始化

RTL8019AS 有四页寄存器组，每页寄存器组有 16 个寄存器。对 RTL8019A 的初始化就是对相关寄存器进行初始化。这些寄存器包括 CR、DCR、RBCR、PSTART、PSTOP、ISR、IMR、PAR0~PAR5、MAR0~MAR5、CURR、TCR 和 RCR 等。初始化过程如下：

- (1) CR=0x21, 选择页 0, 使芯片处于停止模式进行初始化操作;
- (2) TPSR=0x40, 发送缓冲区的起始页地址, 初始化为指向第一个发送缓冲区的页地址为 0x40;
- (3) PSTART=0x4c, PSTOP=0x80, 初始化接收缓冲区的起始页地址和结束页地

址, 构造环形接收缓冲区队列: $0x4c \sim 0x80$;

(4) $BNRY=0x4c$, 设置接收缓冲区读数据指针寄存器使其指向接收缓冲区第一页;

(5) $RCR=0xcc$, 设置接收配置寄存器, 使用接收缓冲区, 仅接收与本机物理地址匹配的数据包以及广播地址数据包和多点播送地址包, 长度小于 64 字节的包丢弃, 校验错的数据包不接收;

(6) $TCR=0xe0$, 设置发送配置寄存器, 启用 CRC 校验码自动生成和自动校验, 工作在正常模式;

(7) $DCR=0xc8$, 设置数据配置寄存器, 使用 FIFO 缓存, 普通模式, 8 位数据 DMA;

(8) $ISR=0xff$, 初始化中断状态寄存器, 清除所有中断状态位;

(9) $IMR=0x00$, 设置中断屏蔽寄存器, 屏蔽所有中断;

(10) $CR=0x61$, 选择页 1 的寄存器;

(11) $CURR=0x4d$, $CURR$ 是 RTL8019AS 写内存的指针, 指向正在写的当前页的下一页, 初始化时指向 $0x4c+1=0x4d$;

(12) 设置多址寄存器 $MAR0 \sim MAR5$, 均设置为 $0x00$;

(13) 设置网卡芯片的物理地址寄存器 $PAR0 \sim PAR5$;

(14) $IMR=0xff$, 重新允许中断;

(15) $CR=0x22$, 选择页 1 的寄存器, 进入正常工作状态;

4.2.2 发送模块

发送模块由函数 `send_frame()` 负责将封装好的数据链路帧写到 RTL8019AS 的指定的发送缓冲区, 然后将数据发送到网络上。

首先判断数据帧长度是否小于 60 字节, 不足 60 字节的要补零, 因为以太网规定数据帧的长度最少 60 字节。然后设定发送缓冲区的起始页和数据长度, 并将需要发送的数据包通过远程 DMA 写入发送缓冲区。最后再根据本地 DMA 将缓冲区的数据发送到网络, 并判断数据发送是否成功, 若失败则可以重发, 总共可以重发 16 次。发送完成后要释放内存。程序流程图如 4-2 所示。

4.2.3 接收模块

接收模块由函数 `rcve_frame()` 负责将 RTL8019AS 接收缓冲区的数据接收到存储

器中。首先判断是否有数据到，即等式 $CURR=BNRY+1$ 是否成立，若不成立则表明有新的数据到来。然后先读取前 4 个字节以获得将要接收的数据长度，再向单片机申请分配已知数据长度大小的内存来存放将要接收的数据。最后将数据读到内存中并返回一个指向该内存缓冲区起始地址的指针。程序流程图如图 4-2 所示。

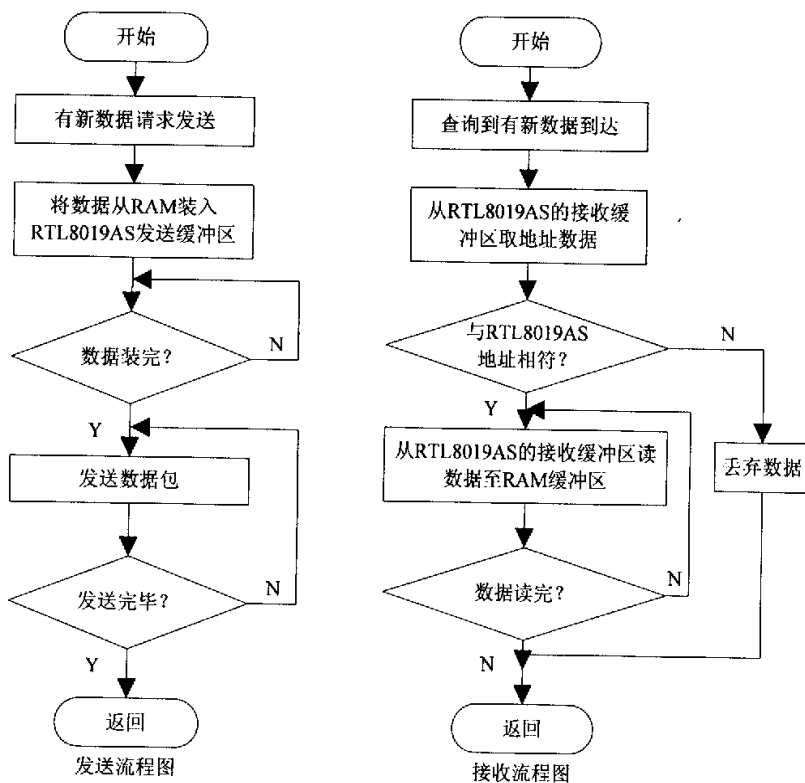


图 4-2 RTL8019AS 接收和发送流程图

4.3 TCP/IP 协议栈及其实现

4.3.1 TCP/IP 协议栈概述

TCP/IP 协议族实际上是许多协议的组合，分为不同的层次，每一层次分别负责不同的通信功能。TCP/IP 协议层次结构和主要的网络协议如表 4-1 所示。

表 4-1 TCP/IP 的协议层次

应用层	Telnet、Ftp、Http 等应用协议
传输层	TCP 和 UDP
网络层	ICMP、IP 和 IGMP
数据链路层	设备驱动程序及接口卡

每一层都负责不同的功能：

(1) 数据链路层通常包括设备的驱动程序和计算机中对应的网络接口卡。它们一起处理与传输媒介有关的物理接口细节。

(2) 网络层负责处理分组在网络中的活动，例如路由的选择。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) 传输层主要为两台主机上的应用程序提供端到端的通讯。在 TCP/IP 协议族中，有两个互不相同的传输协议：TCP 协议(传输控制协议)和 UDP 协议(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。而另一方面，UDP 则为应用层提供一种非常简单的服务。它只是把数据报的分组从一台主机发送到另一台主机，不保证数据报能到达另一端。任何必须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。主要包括 Telnet(远程登录)、FTP(网络传输协议)、SMTP(简单邮件传送协议)、HTTP(超文本传输协议)等协议。

4.3.2 TCP/IP 协议栈的数据封装

当应用程序用 TCP 传送数据时，数据被送入协议栈中，然后逐个通过每一层直到被当作一串比特流送入网络。其中每一层对收到的数据都要增加一些数据报头、数据报尾和其他信息。应用层待发送的数据根据需要确定传输层协议 TCP 或 UDP，然后将数据传送给相应的 TCP 或 UDP 进程，数据在这里进行传输层协议封装(即数据前加上了 TCP 或 UDP 首部)，然后发送给 IP 进程，IP 进程通过 ARP 协议将上层数据报中的 IP 地址转化成网络物理地址(MAC 地址)，同时将 IP 首部置于数据报头部，接下来的工作就由驱动程序和硬件电路完成相应的以太网帧封装，并发送至网络。

由于 TCP、UDP、ICMP 和 IGMP 都要向 IP 传送数据，因此 IP 必须在生成的 IP 首部中加入某种标识，以表明数据属于哪一层。为此，IP 在首部中存入一个长度为 8 比特的数值，称作协议域。1 表示为 ICMP 协议，2 表示为 IGMP 协议，6 表示为 TCP 协议，17 表示为 UDP 协议。

当目的主机收到一个以太网数据帧时，数据就开始从协议栈中由底向上升，同时去掉各层协议加上的报文首部。每层都要去检查报文首部中的协议标识，以确定接收数据的上层协议。如图 4-3 所示，这部分工作是由用户程序来完成的，RTL8019AS 仅负责把数据接受进其内部的数据缓冲区。

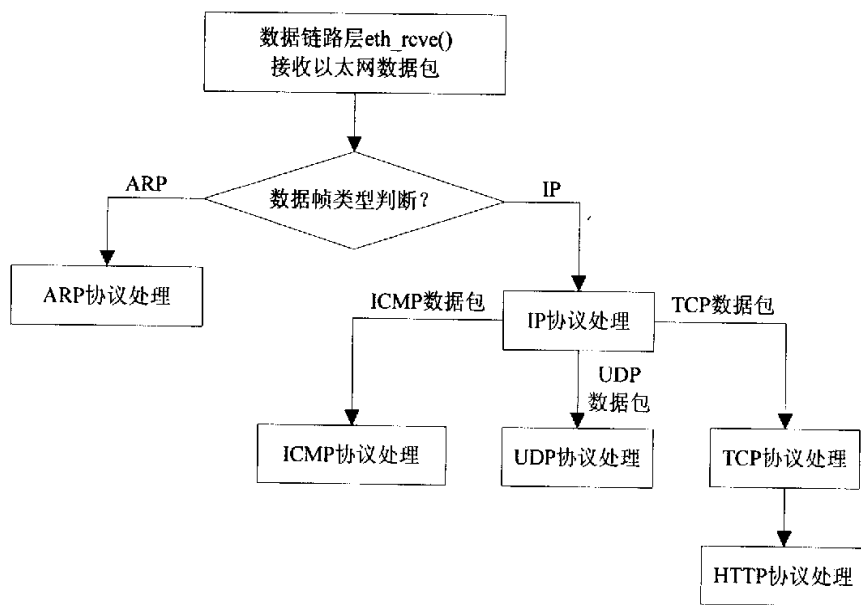


图 4-3 TCP/IP 协议栈工作流程图

4.3.3 TCP/IP 协议栈的裁减

TCP/IP 协议族是一个非常庞大的体系，内容复杂且协议种类繁多，而单片机系统资源十分有限，不可能将所有协议进行集成，只能在对 TCP/IP 进行裁减的基础上，将它作为一种嵌入式应用，固化到单片机系统的 ROM 中，使数据传输按 TCP/IP 协议格式封装，从而达到在 Internet 上传输的目的。对协议的适当简化不但可以节约硬件成本，而且能提高数据吞吐率。因此，本系统在实现 TCP/IP 协议栈时，综合考虑了系统资源和实用性等各种因素后，只设计并实现了 TCP/IP 协议栈中的数据链路层协议 ARP、网络层协议 IP 和 ICMP、传输层协议 TCP 和 UDP 以及应用层协议 HTTP。

4.3.4 TCP/IP 协议栈的实现

4.3.4.1 ARP 软件设计

当源端的 IP 数据报要发送到目的端时，需要将 IP 数据报封装成数据链路帧，这时就需要知道目的端的物理地址，完成查找目的物理地址这一功能就需要 ARP（地

址解析协议)。

ARP 主要完成 IP 逻辑地址到物理地址的转换。IP 地址是二进制 32 位的统一标准的地址，而底层网络的地址长度和格式可能不同，如以太网地址长度为 48 位。当需要将 IP 的数据报传送到目的地时，必须将 32 位的 IP 地址转换成底层网络地址形式。ARP 数据包是固定 28 字节的数据格式，只有“请求”和“回答”两种类型的数据包。它的数据分组格式如图 4-4 所示。

硬件类型 (16bit, 对以太网是 1)		协议类型 (16bit, 对 IPv4 是 0x0800)
硬件长度 (8bit)	协议长度 (8bit)	操作 (16bit, 请求 1, 回答 2)
发送站硬件地址 (对以太网是 6 字节)		
发送站协议地址 (对 IP 是 4 字节)		
目标站硬件地址 (32bit)		
目标协议地址 (32bit)		

图 4-4 ARP 数据分组格式

ARP 软件包划分为三部分：高速缓存控制模块、输出处理模块和输入处理模块。有 6 个函数组成，它们是 ARP 初始化函数 `Init_arp()`、更新 ARP 缓存函数 `age_arp_cache()`、ARP 发送函数 `arp_send()`、ARP 解析函数 `arp_resolve()`、ARP 重传函数 `arp_retransmit()` 和 ARP 接收函数 `arp_rcv()`。

高速缓存控制模块负责维持一个含有物理地址信息的高速缓存表并定期更新表的信息。由于发送站经常与目的站通讯，发送站往往有一个以上的 IP 数据报要发送到同一个目的地。对发送到同一个主机或路由器的每一个数据报都使用 ARP 协议是非常低效的。解决这个问题就是使用高速缓存表。当主机或路由器收到一个 IP 数据报的相应的物理地址时，就可将此物理地址存储在高速缓存表中。对于发往同一个目的端的数据报，这个地址在短时间内还能够使用。但是由于高速缓存表的空间很有限，映射到高速缓存表只能维持有限的时间。

高速缓存表用结构数组来实现，在本设计中，每个项目包含以下一些字段：

- 状态：此 1 字节字段表示项目的状态。它可以是以下 3 个值之一：FREE, PENDING 或 RESOLVED。FREE 状态表示这个项目的生存时间已过期。这个空间可用于新的项目。PENDING 状态表示为这个项目的请求已经发送出，但回答还未收到。RESOLVED 状态表示项目是完整的，已经有了目的地的物理地址。等待发送到这

个目的地的分组可以使用这个项目的信息。

- 尝试：此 1 字节字段表示为这个项目已经发送了多少次的 ARP 请求。若项目状态是 PENDING，则高速缓存控制模块每过一周期就将此字段值加 1，然后再检查尝试字段的值，若此值大于允许尝试则将状态改为 FREE，撤消该项目。若尝试值小于最大数，则创建并发送 ARP 请求报文，直到得到相应的 ARP 回答并获得目的端的物理地址。规定发送请求次数最多为 5 次。

- 协议地址：此 4 字节字段表示目的 IP 地址。

- 物理地址：此 6 字节字段表示目的物理地址。

- 生存周期：此 1 字节字段表示项目的生存时间。每隔一定周期，高速缓存控制模块就将此字段值减去一个常量值，若此时间值为 0 则撤消此项目，将状态置为 FREE。本设计规定每一个项目的生存周期为 5 分钟。

输出处理模块主要由函数 `arp_send()` 和 `arp_resolve()` 组成。`arp_send()` 主要负责创建并发送 ARP 协议的请求或回答报文，具体由传递的函数参数决定。而 `arp_resolve()` 则主要是为 IP 数据报解析目的物理地址。

输出处理模块等待 IP 数据报到达，然后利用掩码运算检查目的 IP 地址是否是本局域网内的 IP 地址，若不是则将默认网关的 IP 地址做为目的 IP 地址，用 ARP 来解析默认网关的物理地址，得到默认网关的物理地址后将 IP 数据报交付到默认网关，由默认网关将 IP 数据报解析并发送到目的地。若检查目的 IP 地址在同一局域网内，则查找高速缓存表，寻找有无一个项目对应于这个数据报的目的 IP 地址。若找到这样的项目，而项目的状态是 RESOLVED，则将该数据报连同目的物理地址一起发送到数据链路层来传输；若找到这样的项目，而项目的状态是 PENDING，则该数据报必须等待，并将数据报存入等待缓冲区，直到找到目的物理地址；若找不到这样的项目，模块就创建一个新项目，然后发送 ARP 请求报文。

输入处理模块负责接收 ARP 分组并根据需要更新高速缓存表，由函数 `arp_rcv()` 获得数据包并进行 ARP 解析。首先判断 ARP 分组首部中的硬件类型和协议类型字段是否符合以太网的定义（硬件类型为 1，协议类型为 4），如果符合则检查 ARP 分组的目的 IP 地址是否是本机 IP 地址，如果目的 IP 地址就是本机的 IP 地址则进一步解析判断 ARP 的操作类型（请求 1，回答 2），若判断条件都不满足则丢弃数据包，程序结束返回。如果 ARP 分组是一个请求报文则用 `arp_send()` 函数创建一个回答报文，在 ARP 分组中填入本机的物理地址，并将操作类型字段设为回答发送给请求端。

这样就完成了 ARP 协议对等进程的一次对话；如果 ARP 组是一个回答报文，那么就在高速缓存表中查找相应的待请求的项目，若找到而项目的状态是 PENDING，则更新该项目，即将分组中的目标物理地址复制到相应的物理地址字段，并将状态改变为 RESOLVED，将尝试字段的值清零，设置这个项目的生存周期为初始值。如果有等待发送的 IP 数据报，就将数据报提取出来，连同其物理地址一起交付给函数 ip_send() 来传输；若找到而项目的状态是 RESOLVED，则仍然更新该项目。这是因为目标物理地址可能已经改变了。生存周期字段的值也要重置；若找不到相应的项目，就创建一个新的项目，并将其增加到这个表中，保留为以后使用。

ARP 软件设计流程图如图 4-5 所示。

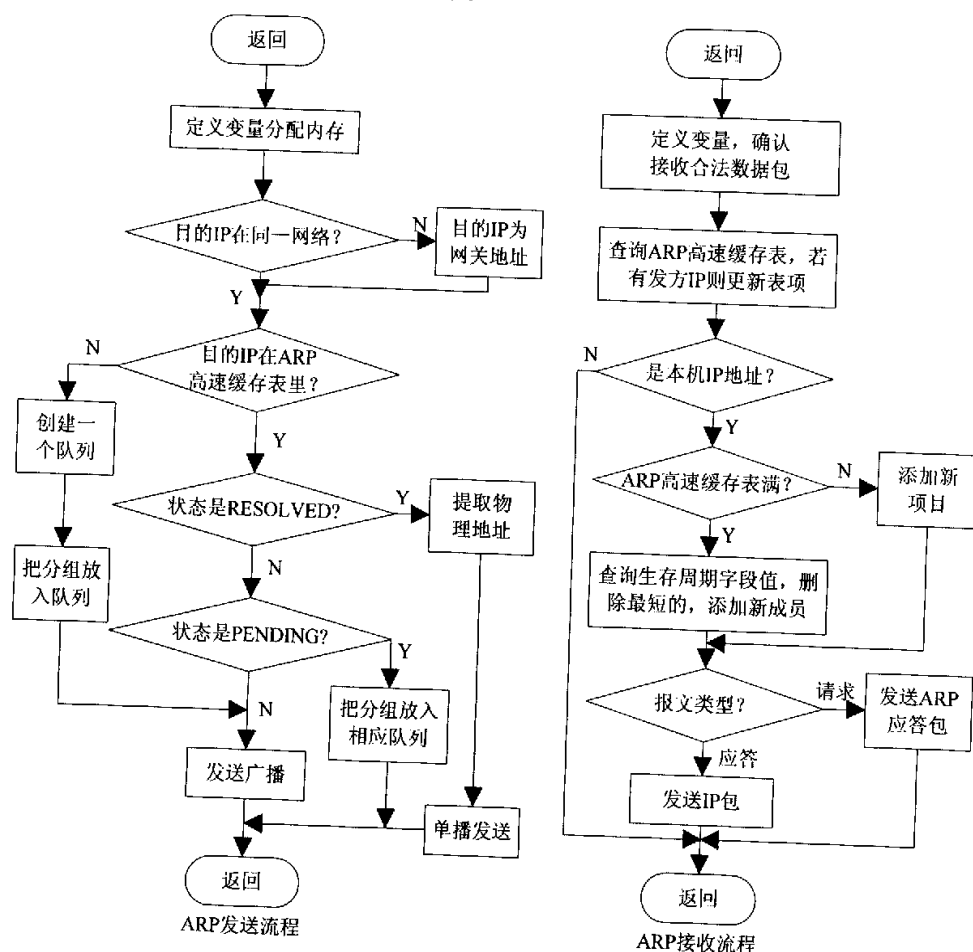


图 4-5 ARP 软件流程图

4.3.4.2 IP 软件设计

IP 协议是一种不可靠的无连接数据报协议，即 IP 不提供差错纠正或跟踪。如

果两个系统连接到同一条链路上，那么通常就不需要网络层，由数据链路层实现的物理编址就可以完成寻址。但是，如果两个系统连接在不同的网络上，而这些网络是由一些连接的设备连接起来，那么通常是需要网络层来完成从源站到目的站的交付，因此，在网络层就需要另一种编址系统来区分离源系统和目的系统，这种逻辑地址称为 IP 地址，它在网络上唯一的。

在 IP 层的分组叫做数据报，它由两部分组成：首部和数据。格式如图 4-6 所示。

VER (4bit)	HLEN(4bit)	服务类型 (8bit)	总长度 (16bit)	
标示 (16bit)			标志(3bit)	分片偏移 (13bit)
生存时间 (8bit)	协议 (8bit)		校验和 (16bit)	
源 IP 地址 (32bit)				
目的 IP 地址 (32bit)				
选项				

图 4-6 IP 数据报格式

以太网数据链路层的 MTU (最大传送单元) 为 1500 字节，而在家电控制中每次传输的数据量最多在几十字节，一般不会超过 1500 字节，因此可以置位 IP 数据报首部中的“不分片”位，将数据报当作只有一个分片来处理以简化对分片数据的处理。另外，IP 数据报的首部包括 20 字节基本部分和 40 字节选项部分。省略选项部分可以提高通信效率，因此只保留 20 字节的基本首部。

IP 软件包包括输入处理模块和输出处理模块，有两个函数组成：IP 接收 `ip_rcve()` 和 IP 发送 `ip_send()`。

输入处理模块主要由函数 `ip_rcve()` 完成。首先判断接收数据包的目的地址是否是本机的 IP 地址，如果是则计算数据包的检验和，否则丢弃数据包，程序结束返回。如果检验和不是 0 则表明数据包传输有误，丢弃数据包，程序结束返回。如果检验和计算正确，则判断 IP 数据报首部中版本号是否是 4 以及本数据报是否是一个分片。若 IP 数据报首部的版本不是 4 则把数据报当作错误数据报处理，不予解析。另外，本 IP 协议规定不处理分片数据报，因此若是分片数据报则丢弃。最后，如果 IP 数据报首部含有选项部分则调整数据报首部，将超过 20 字节的选项部

分删除，只解析固定首部的 20 字节。若上述解析正确无误，就提取 IP 数据报的数据部分，并根据使用 IP 协议的上层传输层协议将数据部分传递到上层相应的协议进一步解析。

输出处理模块主要由函数 `ip_send()` 完成。当上层协议有数据要发送时，就将数据传递到 IP 的输出处理模块。函数 `ip_send()` 首先将上层数据封装成 IP 数据报，即添加 IP 数据报的首部并计算校验和。然后调用 ARP 模块的物理地址解析函数 `arp_resolve()` 获得目的物理地址。若得到目的物理地址则将 IP 数据报向下发送到数据链路层，若没有获得则等待 ARP 模块解析。

IP 软件设计流程图如图 4-7 所示。

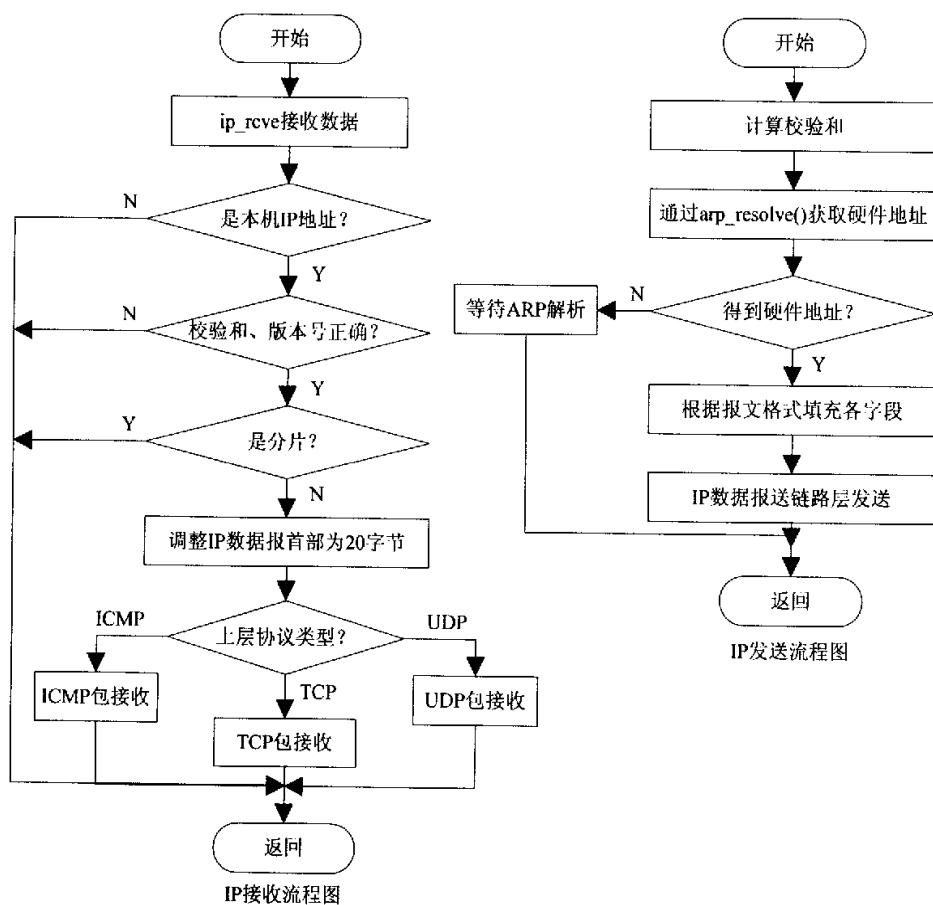


图 4-7 IP 软件流程图

4.3.4.3 ICMP 软件设计

由于 IP 协议软件设计是一个不可靠的、无连接的协议，因此需要 ICMP 的配合

以弥补 IP 缺少差错控制和辅助机制的缺点。ICMP 报文分为差错报告报文和查询报文两大类。

差错报告报文报告路由器或主机(目的站)在处理一个 IP 数据报时可能遇到的一些问题。所有的差错报文都包括一个数据部分,而这数据部分包括原始数据报的首部加上数据报中的前 8 个字节的的数据,加上原始数据报中的首部就可给出原始信源,它接收差错报文,这是数据报本身的信息。要包括数据的前 8 个字节是因为这前 8 个字节提供了关于端口号(UDP 和 TCP)和序号(TCP)的信息,这些信息是需要的,因而源站可以将差错情况通知这些协议(TCP 或 UDP),ICMP 形成差错分组,然后再封装成 IP 数据报。

查询报文是成对出现的,它帮助主机或网络管理员从一个路由器或另一个主机得到特定的信息。

本模块由三个函数组成: Ping 发送 ping_send()、目的站不可达发送 dest_unreach_send()和 ICMP 报文接收 icmp_rcve()。

Ping 发送函数主要是测试发送方和接收方是否正确连接,在 IP 层是否能够通信;Ping 发送的报文就是接收的报文,它把接收的数据又发送回去,是测试主机的可达性,不实现别的功能。

ICMP 目的站不可达函数主要是处理当一个 UDP 报文没有到达目的端口,主机就发送一个不可达报文,通知发送主机,发送失败。

ICMP 接收函数主要处理两种类型的 ICMP 报文,一种是目的不可答,另一种是 Ping 请求接收,除了这两种类型,别的类型不识别。

ICMP 软件设计流程图如图 4-8 所示。

4.3.4.4 UDP 软件设计

UDP 是一种简单的传输层协议,处于应用层和网络层之间,为应用层提供一种比较简单的服务,是一种无连接的、不可靠的传输层协议。UDP 使用端口号创建进程到进程的通信,不提供流控制,只提供程度很弱的差错控制。

UDP 叫做用户数据报,有 8 个字节的固定首部。用户数据报格式如图 4-9 所示。

源端口号 (16bit)	目的端口号 (16bit)
总长度 (16bit)	校验和 (16bit)
数据	

图 4-9 用户数据报格式

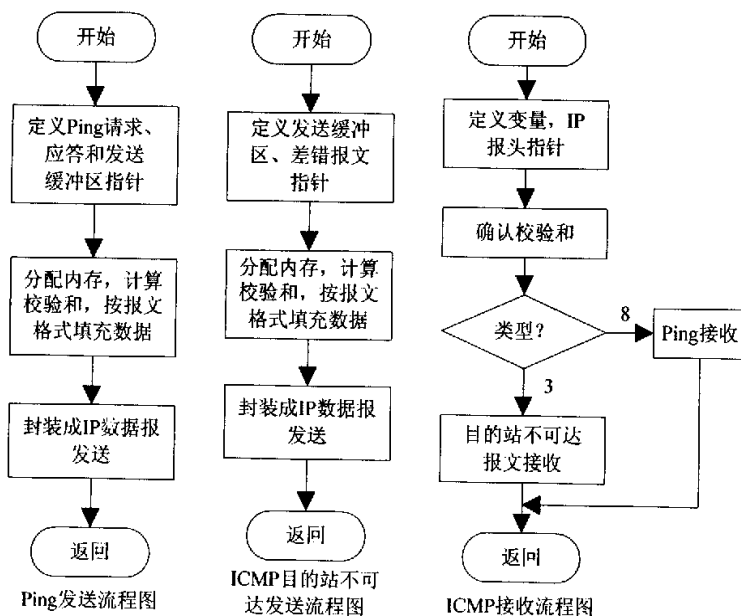


图 4-8 ICMP 软件流程图

UDP 软件包包含输入处理模块和输出处理模块，有三个函数组成：UDP 回应 `udp_echo_service()`、UDP 接收 `udp_rcve()` 和 UDP 发送 `udp_send()`。

输入处理模块调用 `udp_rcve()` 从 IP 协议接收用户数据报，进行校验和运算，然后根据 UDP 用户数据报首部中目的端口号字段的值把数据报中的有效数据再递交到应用层协议，由 `udp_echo_service()` 作进一步处理；

输出处理模块有 `udp_send()` 完成，负责接收从应用层协议传递的数据并添加 UDP 首部，发送用户数据报到网络层的 IP 协议。

UDP 软件设计流程图如图 4-10 所示。

4.3.4.5 TCP 软件设计

TCP 是整个 TCP/IP 协议族的核心，也是传输层中最复杂的协议。TCP 提供的服务包括：（1）流式数据服务，TCP 从发送应用程序接收到字符流，从这个流中提取适当的长度创建叫做报文段的分组，然后将它们发送到网络上。（2）全双工服务，即数据可在同一时间双向流动。（3）可靠服务，使用确认机制和差错控制来实现数据完整和安全的传递。

TCP 中的差错控制通过三种工具来完成：（1）校验和。（2）使用确认的方法来证实收到了某些报文段，它们已经无损失的到达了目的地。（3）若一个报文段在超时时间截止之前未被确认，则被认为是受到损伤或丢失。

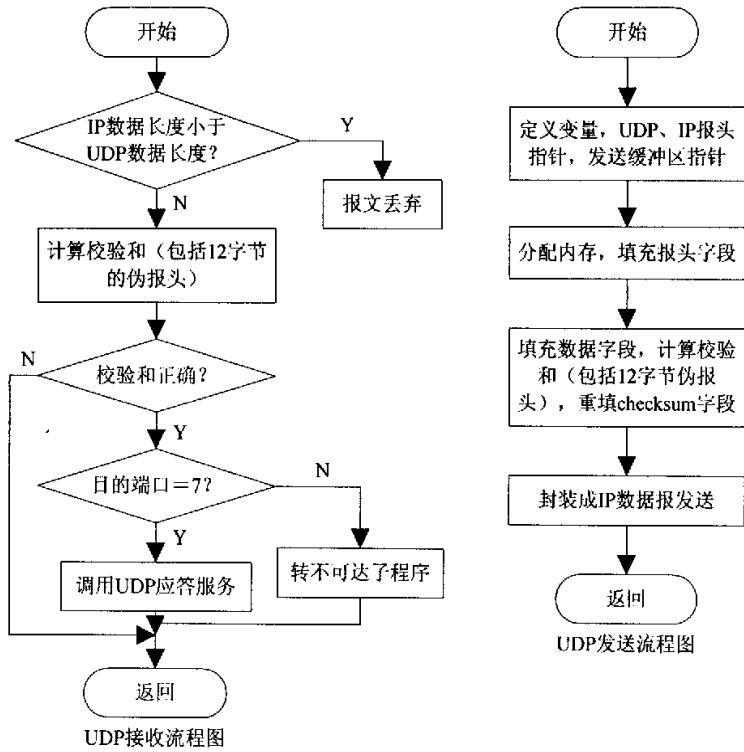


图 4-10 UDP 软件流程图

TCP 协议是一个面向连接的协议，在源站和目的站之间建立了一条虚路径，属于一个报文的所有报文段都沿着这条虚路径发送，整个报文使用一条虚路径使得确认过程以及对损伤和丢失报文的重传更加容易。在 TCP 中，面向连接的传输是通过两个过程来完成的：连接建立和连接终止。建立一个连接需要三次握手，而终止一个连接需要经过四次握手。

两个通信端使用 TCP 传送的数据单元叫做报文段，其数据格式如图 4-11 所示。

源端口地址 (16bit)				目的端口地址 (16bit)				
序号 (32bit)								
确认号 (32bit)								
首部长度 (4bit)	保留 (6bit)	urg	ack	psh	rst	syn	Fin	窗口大小 (16bit)
校验和 (16bit)						紧急指针 (16bit0)		
选项及填充								

图 4-11 TCP 报文段格式

TCP 是一个机制非常复杂的协议，一个完整的 TCP 软件包通常需要代码多达几百万行。在 PC 机中一般是以多个任务或进程实现的，每一个状态就是一个任务或进程。而在嵌入式系统中，各种系统资源都十分有限，不可能像 PC 机那样有足够的条件，因此在设计 TCP 软件时，应根据系统的实际需要，对原有的 TCP 协议进行适当的简化和改写。

(1)、简化 TCP 状态机

TCP 数据传输可分为三个阶段：建立连接、传输数据和断开连接。它的实现过程可用状态机来描述。为了保证状态机的正确性，在其实现的时候，不同状态的相关信息需要存储下来。状态机越复杂，维护的花费就越大，对处理器的存储能力和运算能力的要求就越高。标准的 TCP 状态机如图 4-12 所示^[20]，图上每一条线都注上用斜杠隔开的两个字符串，第一个字符串是输入，即 TCP 所收到的。第二个字符串是输出，即 TCP 所发送的；虚线表示服务器的状态变迁图，实线表示客户的状态变迁图，而较细的线对应用非正常情况。从图中可以看出，要维护这样的一个庞大的状态机，就低速处理器的存储能力和运算能力来说，是十分困难的。

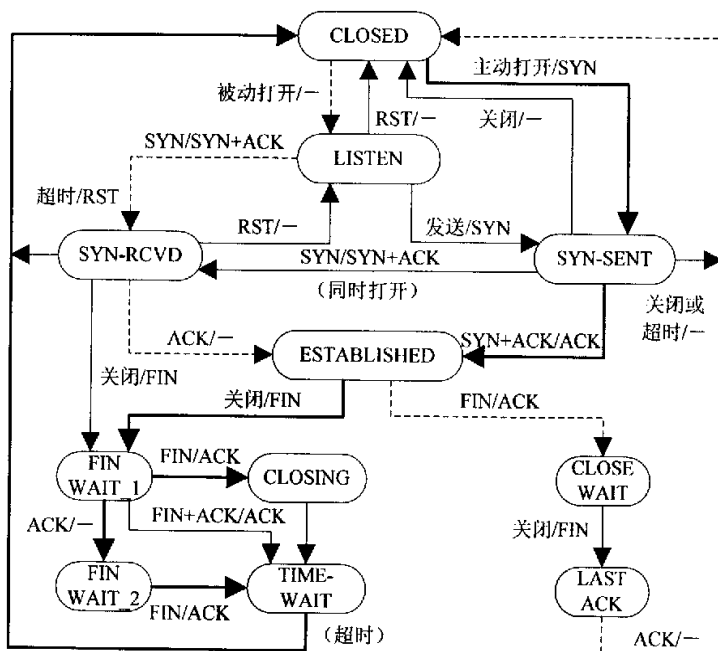


图 4-12 TCP 状态转化图

图中 ESTABLISHED 状态之上的几个状态属于建立连接部分，状态之下的几个状态属于断开连接部分。建立连接有主动打开和被动打开两种方式。服务器是一种被

动打开，它一直在侦听连接请求；客户端是一种主动打开，它发送连接请求以建立连接。从初始的关闭状态 CLOSED，到连接建立状态 ESTABLISHED，基本是沿两条路线进行，左边一条是服务器端建立连接的过程，经历了 CLOSED→LISTEN→SYN_RCVD→ESTABLISHED 四个状态的转换。右边一条是客户端建立连接的过程，经历了 CLOSED→SYN_SEND→ESTABLISHED 三个状态的转换。基于对家电控制的特定应用，为了实现一个无冗余的系统，在 TCP 层的上层实现的是服务器端的应用，可以将标准 TCP 状态机建立连接过程中客户端建立连接的状态机部分简化掉。

断开连接有两种方式，一种是主动断开连接，另一种是被动断开连接。被动断开连接的处理较为简单，但经常需要实现主动断开连接，特别当上层应用实现的是服务器端功能时，从安全的角度考虑，希望服务器端能够实现主动断开连接，但标准 TCP 协议的主动断开连接的状态机部分过于复杂。经实验发现，在需要主动断开连接时，发送一个 FIN 数据包，在接收到 FIN 数据包的确认后再发送一个 RESET 数据包，就可顺利完成一次主动断开连接。

(2) 使用简单的确认机制

序列号和确认号这两个字段用于协同完成 TCP 协议中的确认工作。对于每个接收到的数据包进行确认号的计算，需要从接收到的数据包中提取 TCP 报文的数据部分长度，并进行计算，这增加了处理器的运算量。但如果每次只对单个 TCP 报文进行确认的话，并没有太大的难度。问题在于标准的 TCP 协议使用滑动窗口机制。滑动窗口是一种在流量控制和网络传输效率之间取得折中的方案，它允许发送方在等待一个确认之前发送多个窗口。对于使用滑动窗口的 TCP 连接，其确认是一种批量的确认，处理器要对多个数据包连续传输进行维护和处理，困难较大。在信息家电控制中，每次传输的数据量不大，并且嵌入式系统中的接收缓存也是固定大小，因此将 TCP 首部中窗口字段设定为固定大小的长度，不必每次变换窗口大小。

TCP 软件包主要包括传输控制块、输入处理模块、输出处理模块和计时器模块。由 5 个函数组成：TCP 初始化 `init_tcp()`、TCP 保活 `tcp_inactivity()`、TCP 接收 `tcp_rcve()`、TCP 发送 `tcp_send()` 和 TCP 重传 `tcp_retransmit()`。

传输控制块以一个表的形式记录维持 TCP 连接的状态信息，通常是结构数组。

输入处理模块是 TCP 软件中最重要的组成部分，它的主要任务是调用 `tcp_rcve()` 接收网络层 IP 协议传送的数据，并对有效的数据部分进行 TCP 协议解析处理，并判断数据分组的下一步流向，进而将数据送交到不同的上层协议处理。

输出处理模块任务比较简单，主要完成 TCP 协议的数据发送，即将需要发送的数据封装成 TCP 协议数据格式，调用 `tcp_send()` 发送到下层 IP 协议。

计时器模块主要是为 TCP 软件提供一组计时功能，包括重传计时器、保活计时器。重传计时是 TCP 的一种差错控制机制，当 TCP 发送报文段时，它就创建该报文段的重传计时器。若在计时器截至时间到之前收到了对此报文段的确认，则复位此计时器，若在收到对此报文段的确认之前计时器截止时间到，那么就认为此报文段已丢失或出现差错，重传此报文段。保活计时器是用来防止两个 TCP 之间的连接处于长时间空闲，当 TCP 连接空闲超过规定时间后就终止该 TCP 连接。

TCP 软件设计主要工作流程如图 4-13 所示。

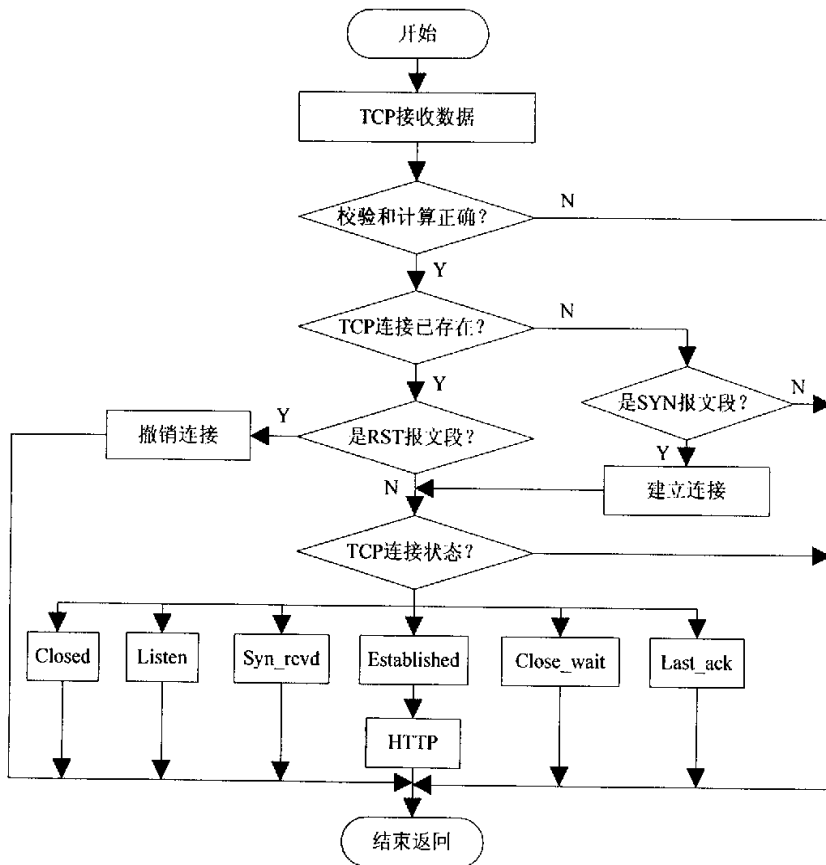


图 4-13 TCP 软件流程图

4.3.4.6 HTTP 软件设计

HTTP 是 TCP/IP 协议栈的应用层协议，它主要是给用户提供了与 TCP/IP 协议的接口，是互联网上存取数据的协议，使用传输层的 TCP 通用端口号 80。HTTP 协议

的思想很简单，即客户端发送一个请求，服务器端创建一个回答，因此它只有两种类型的报文：请求报文和响应报文。

请求报文的数据格式包括一个请求行、一个首部以及有时出现一个主体。请求行定义请求类型、资源（URL）和 HTTP 版本；首部是在客户端和服务器端交换数据所附加的信息，它指明客户端的配置信息和使用的文档格式。

响应报文的数据格式包括一个状态行、一个首部及响应的数据。状态行定义响应的状态，包括 HTTP 版本、状态码、状态短语；首部指明服务器的配置和关于请求的特殊信息。

HTTP 软件包由两个函数组成：`http_server()` 服务器模块和 `http_send()` 发送模块。

`http_server()` 服务器模块主要是接收下层 TCP 协议传送的有效数据，解析请求端发送过来的命令，如果是命令“GET”，继续扫描后续字符，把数据存储在单片机存储单元中，并创建一个响应，调用 `http_send()` 将数据发送出去。

HTTP 软件设计流程图如图 4-14 所示。

4.4 协议转换程序

（1）以太网转换到 RS232 端口的发送程序

将 TCP/IP 协议的数据发送到 RS232 端口的处理中，针对数据包的不同类型作不同的处理。如果是 ARP 包或 ICMP 包，则不作处理直接返回，如果是 TCP 包，则将 TCP 包去头取数据，将原始变换为 RS232 格式的数据向相应的端口发出去。

（2）RS232 转换到以太网的接收程序

在对 RS232 的数据的处理过程中，首先从 RS232 端口的缓冲区里取出数据，然后依次按层加上相应层的协议头后，通过以太网发送给服务器。

协议转换程序流程图如图 4-15 所示。

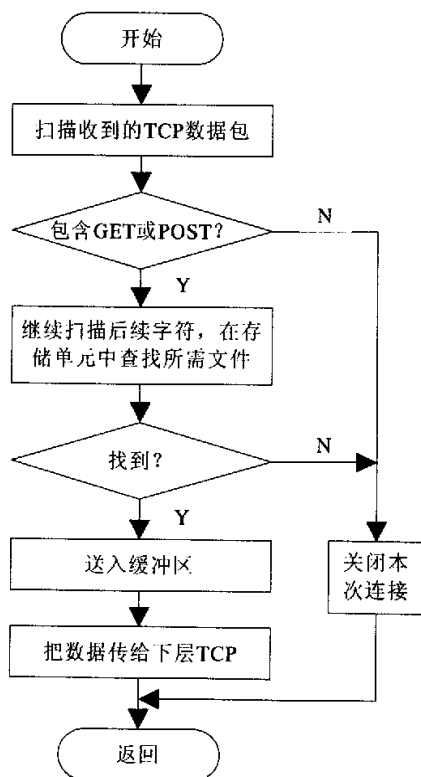


图4-14 HTTP软件流程图

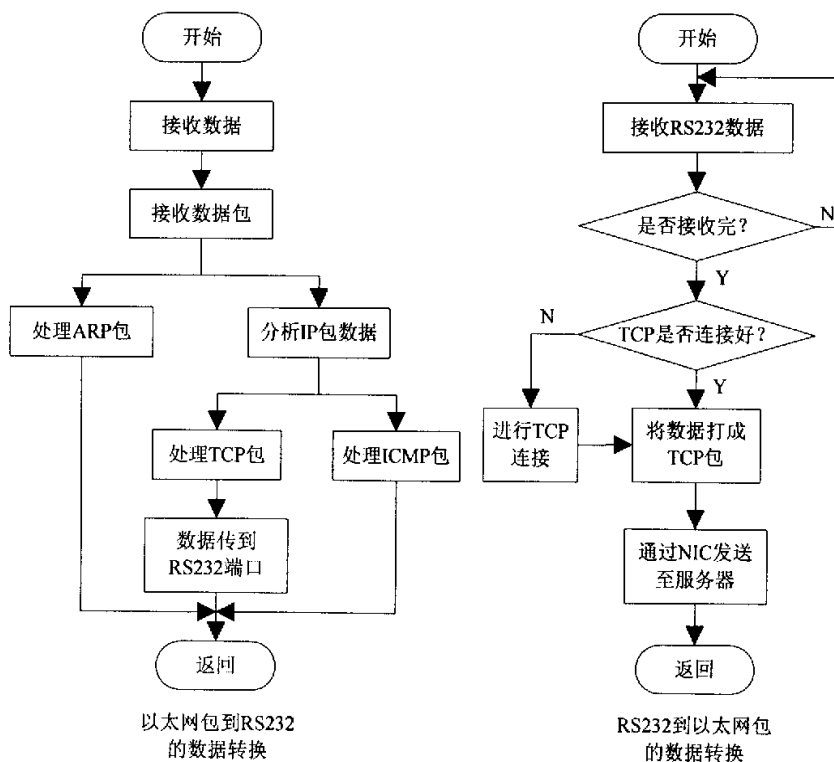


图 4-15 协议转换程序流程图

(3) 串行通信协议

把串行口工作波特率设置为 9600，8bit 数据位，1bit 停止位，无校验位，无流控制。嵌入式网络接口模块向家电控制器发送控制命令，然后接收家电控制器传送回来的信息转换成 web 页面。

以空调控制器与嵌入式网络接口模块之间的通信为例，通信双方所传送的主要有开关机、当前室温、设定温度、工作模式、定时、风速和类型以及空调的状态等参数。参数如表 4-2 所示。

表 4-2 嵌入式网络接口与空调控制器通信协议

项目名称	串口发送数据	串口接收返回数据
取得空调工作参数	G	0/1: power on/off
		0/1: 是否定时
		0~9: 定时时间
		1~4: 工作模式
		2~4: 风扇风速

		0~99: 目前房间的温度
		0~99: 目前设定的温度
开机	K	k
关机	S	s
增温	A	a
减温	D	d
增加定时时间	I	i
减少定时时间	J	j
工作模式	U	u: STOP 模式
	V	v: AUTO 模式
	X	x: FREEZE 模式
	Y	y: DRY 模式
	Z	z: WIND 模式
风速	O	o: 1 速
	P	p: 2 速
	Q	q: 3 速
	T	t: 0 速

4.5 本章小结

本章按照系统设计的要求,对 TCP/IP 协议栈进行了精简。实现了以太网控制器的驱动,裁减了 TCP/IP 协议,实现了 ARP、IP、ICMP、UDP、TCP 和 HTTP 等协议,以及嵌入式网络接口模块与家电控制器的协议转换程序,并给出了相应的程序流程图。

第5章 家电控制器的设计与实现

数字家庭一个主要应用领域就是基于网络实现对家电的监控,如通过 Internet 对家里的空调进行遥控操作,包括开机、关机、温度调节等。家电除了要完成常规的功能以外,还要和外部网络进行通信,接收远程用户通过 Internet 发送过来的指令,分析指令并执行相应的操作,根据需要返回家电的工作状态参数。在本章中详细的介绍了电风扇控制器和空调控制器的软硬件设计,控制器通过串口与嵌入式网络接口模块通讯,使电器接入 Internet,完成和外部网络的通信。

5.1 电风扇控制器的设计

5.1.1 控制器的性能指标要求

设计的电风扇比普通电风扇功能强大,操作简单化、智能化、网络化,主要实现以下几个部分的功能:

(1) 温度智控功能: 风扇可以感知环境的温度,以调节风扇的转速,达到更好的工作效果。用户可以选择这种智能调速方式,也可以选择手动设定方式来控制转速。当选择手动设定方式时,该功能不发挥作用。

(2) 多级调速功能: 提供更多的风力级别和风型,提高用户的舒适度。

(3) 定时工作功能: 该定时功能可以让用户自己定制风扇工作时间的长短,以提供更人性化的服务。

(4) 数码显示功能: 使用数码管显示当前室温,风扇的转速,风扇的定时时间等参数。

(5) 红外遥控功能: 提供远距离非接触式的风扇控制操作。

(6) 通讯功能: 与嵌入式网络接口模块进行通讯,使电风扇接入 Internet。

5.1.2 控制器的硬件组成

电风扇控制器的硬件原理框图如图 5-1 所示。由图可知,电风扇控制器的硬件是以 AT89S52 微处理器为核心,主要由电源电路、温度检测电路、面板按键、红外接收电路、数码显示、串行通信接口、电机控制与驱动电路和报警提示电路等几部分组成。

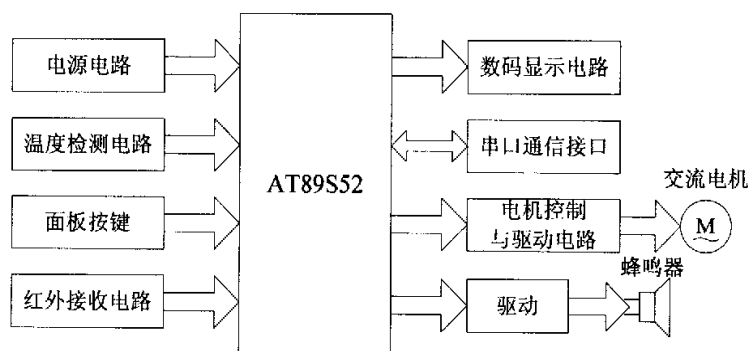


图 5-1 电风扇控制器硬件系统框图

5.1.3 微处理器的选择及特点

微处理器是电风扇的核心，它响应用户对电风扇的直接操作，控制各个模块的工作，对各器件进行有序控制；通过通信模块发送信息到主网络，并接收从通信模块传来的信息，对各种控制、查询命令做出响应，提供用户远程监控功能。

随着微处理器应用的日益广泛，种类越来越多，一种微控制器的选择需要从多方面考虑，首先，必须根据系统要求，在满足功能要求前提下，优化系统的构成；其次，要想使用某种微控制器设计好一个系统，必须对这种微控制器及其系统比较熟悉；最后，开发设备和产品的供货渠道以及成本也确定能否采用某种产品。

在电风扇开发领域，业界采用的大都是专用的风扇控制芯片，它实现简单，但是跟本论文设计的要求相比，其处理计算能力就相差甚远。在认真分析了功能需求和成本的基础上，决定选用 AT89S52 作为本系统的核心控制部分，主要原因是：

(1) 它是一种低功耗、高性能 CMOS 8 位微控制器，与工业 80C51 产品指令和引脚完全兼容。这对于熟悉 MCS51 系列的用户来说用 AT89 系列单片机取代 51 系列进行系统设计是轻而易举的。

(2) 具有在系统可编程 Flash 存储器，使用 Atmel 公司高密度非易失性存储器技术制造。这在系统的开发过程中，可随意进行程序修改，方便了产品开发且大大缩短了程序的开发周期。8K 字节 Flash 完全可以满足系统程序的存储容量，不必再扩展系统；芯片内含 256 字节的 RAM，也可以省去 RAM 的扩展。

(3) 具有丰富的片内资源：32 位 I/O 口线，看门狗定时器，3 个 16 位定时器/计数器，一个 6 向量 2 级中断结构，全双工串行口，片内晶振及时钟电路等。

(4) 芯片工艺成熟，从已使用的情况看芯片的 EMC 性能很好，并且价格相对比较低廉。

总之，AT89S52 具有丰富的资源和相对低廉的价格，是一种较好的选择。

5.1.4 电源电路

电路板上需要两种电压：AC220V 和 DC5V。AC220V 直接给电风扇和转页供电；DC5V 用于微控系统供电。电源电路如图 5-2 所示，电源采用电容降压方式，缩小了电路板的体积，最大能提供 50mA 的电流。由于测温部分要求电压的稳定精度高，所以在稳压管 D7 稳压后使用了三端稳压管 7805 将电压稳压在+5V，R10 用于泄放 C8 上的电压，R11 可减少接通电源瞬间对 D7 的冲击。

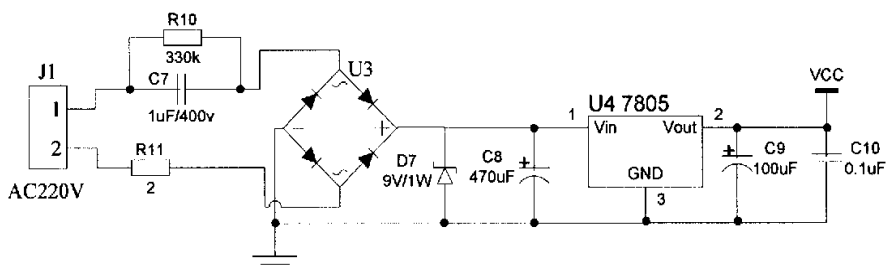


图 5-2 电源电路

5.1.5 温度检测电路

测温采用普通的二极管 1N4148 作为温度传感器，利用二极管两端的电压与温度成反比的特性检测室温。当流过二极管的电流恒定时，二极管两端的电压与温度成反比，在 $-55^{\circ}\text{C} \sim 125^{\circ}\text{C}$ 有着良好的线性，温度每上升 1°C ，电压大约下降 2.18mV 。测温电路如图 5-3 所示。

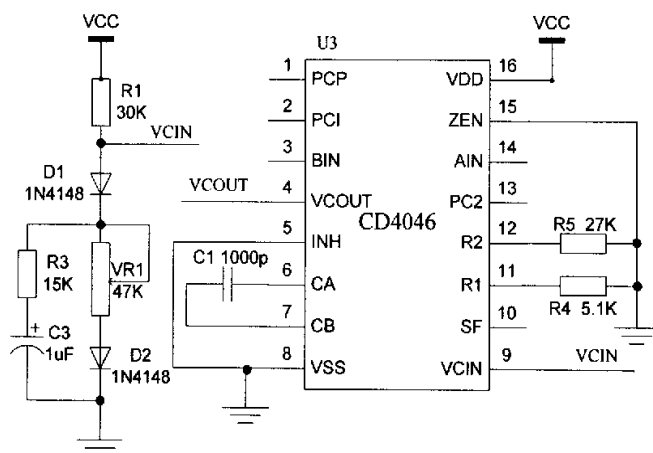


图 5-3 测温电路

测温电路采用一片锁相环集成电路 CD4046，只使用其电压/频率变换部分，它

的输出频率与输入电压成正比,比例与图中的 R4、R5、C1 有关。当输入电压在 1.1~3.2V 范围内时输出频率的线性好,由于二极管工作时其两端的电压不到 1V,故串接电阻 R3 把输入电压抬高,CD4046 的频率输出脚 VCOUT 接到单片机的 T0 (P3.4 脚)。

测温时把单片机的 T0 设为 16 位计数方式, T1 设为 16 位定时方式,定时时间为 100ms,测温过程如下:开 T1 中断,启动定时器 T1,同时启动计数器 T0,当 T1 定时时间到后,进入 T1 中断服务程序,停止 T0 计数,所测频率 $f = \text{计数值} / 0.1s$,温度 = $(f_0 - f) / k$, f_0 为 0℃ 时的输出频率, k 为输入电压变化 2.18mV 对应的频率的变化值 (100Hz 左右)。

5.1.6 人机接口电路

在单片机的应用系统中,通常都要有人机对话功能。它包括人对应用系统的状态干预与数据输入以及应用系统向人报告运行状态与运行结果。本设计选择键盘和红外遥控作为操作者对应用系统的状态干预与数据输入的外部设备,用 LED 显示器和蜂鸣器来向操作者报告系统运行状态与运行结果。

5.1.6.1 按键接口电路

按键在单片机应用系统中是一个关键部件,它可以实现向单片机输入数据、传送命令等功能,是人工干预的重要手段。在单片机组成的测控系统中用得最多的是独立联接式和矩阵式两类。由于只需要调档和定时两个按键,按键较少,所以本设计就采用了独立联接式设计按键。在主程序中调用键盘扫描程序,可测出两个键中哪个键被按下,当测试到某个键按下时设置了 20ms 的去抖动时间,然后才读取键值。只有当按下的键释放后才开始新一轮的扫描,这样的方法可以防止按键的误操作。

5.1.6.2 红外接收电路

红外遥控器的输出是对编码后的串行数据用 38~40KHz 的方波进行脉冲幅度调制而产生的,所以解码时要先对已调制的波形进行解调,然后对解调后的波形进行解码。接收电路可以使用一体化红外线接收器成品。接收器包括红外接收管和信号处理 IC,首先将遥控

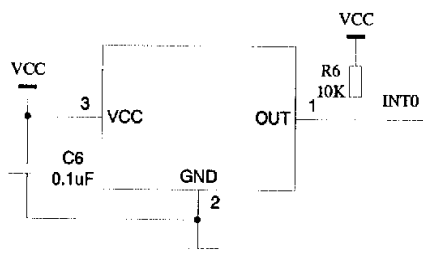


图5-4 红外接收电路

器来的光信号转换成电信号，再经前置放大器、包络检波、整形以后转换为微处理器可以识别的 TTL 电平信号，然后微处理器对 TTL 电平信号序列进行读取解码。接收器有 3 个引脚：Vcc、GND 和一个脉冲信号输出 OUT，该输出信号被连接到单片机的外部中断 0 输入端，并通过 10k 的上拉电阻连接到电源 VCC，如图 5-4 所示。

5.1.6.3 LED 显示电路

LED 显示电路主要用于显示当前环境温度、风扇档位和定时时间。电路采用了三位数码管，动态扫描方式，占用 11 根 I/O 线。P0 口输出显示代码，P2.2、P2.3 和 P2.4 口提供位扫描，动态驱动相应的 LED。由于单片机电流驱动能力较小，为此，使用了驱动三极管 PSS9015 作为地址驱动。为了能显示“℃”，把数码管倒着安装，温度显示为 xx℃，风扇档位显示为 F = x，定时时间显示为 H = x，关机显示为 OFF。

5.1.7 串行通信接口电路

串行通信接口电路是连接电风扇与嵌入式网络接口模块的桥梁。AT89S52 内带有一个可编程的全双工串行接口，该串行接口具有多级通信功能，可以与 PC 机、数据收发设备或其他单片机构成通信系统。其接口电路原理图如图 3-4 所示。

5.1.8 电机控制电路

电机调速是整个控制系统中一个重要的方面，通过控制双向可控硅的导通角，使输出端电压发生改变，从而使施加在电风扇的输入电压发生变化，调节风扇的转速，实现自然风、睡眠风的无级调速。

本设计采用过零触发双硅输出光耦 MOC3061，驱动可控硅 TLC336A 来控制电风扇叶，电机控制与驱动电路如图 5-5 所示，从 CPU 来的控制信号经 MOC3061 耦合后，接可控硅 TLC336A 的门极，以控制电机的运转。采用过零触发可减少风扇的噪声、减少干扰且低速档启动性能好。MOC3021 是普通的可控硅输出光耦，直接控制转页。为了提高电路的可靠性，并入 RC 串联电路实现对电弧吸收。

可控硅的导通条件如下：1. 阳-阴极间加正向电压；2. 控制极-阴极间加正向触发电压；3. 阳极电流 I_A 大于可控硅的最小维持电流 I_H 。

电风扇的风速分为高、中、低三档，各档的风速都有一个限定值。在额定电压、额定功率下，以最高转速运转时，要求风叶最大圆周上的线速度不大于 2150 m/min，且线速度可由公式 $V = \pi Dn \times 10^{-3}$ 求得。式中：V 为扇叶最大圆周上的线速度 (m/

min), D 为扇中的最大顶端扫出圆的直径(mm); n 为电风扇的最高转速 (r/min)。代入数值可求得, $n_{高} \leq 1555$ r/min, 取 $n_{高} = 1250$ r/min。又因为调速比=最低转速档的转速/最高转速档的转速 $\times 100\% \leq 70\%$, 取 $n_{低} = 875$ r/min, 由此可得, $n_{中} = 1063$ r/min。又由于负载上电压的有效值 $u_0 = u_1 (\sin 2\alpha / (2\pi) + (\pi - \alpha) / \pi)^{1/2}$ 。其中: u_1 为输入交流电压的有效值; α 为控制角。解得 $\alpha_{高} = 0^\circ$, $t = 0\text{ms}$; $\alpha_{中} = 46.5^\circ$, $t = 2.58\text{ms}$; $\alpha_{低} = 76.5^\circ$, $t = 4.25\text{ms}$ 。

以上计算出的是控制角和触发时间,当检测到过零点时,按照所求得的触发时间延时发送脉冲,便可实现预期转速。

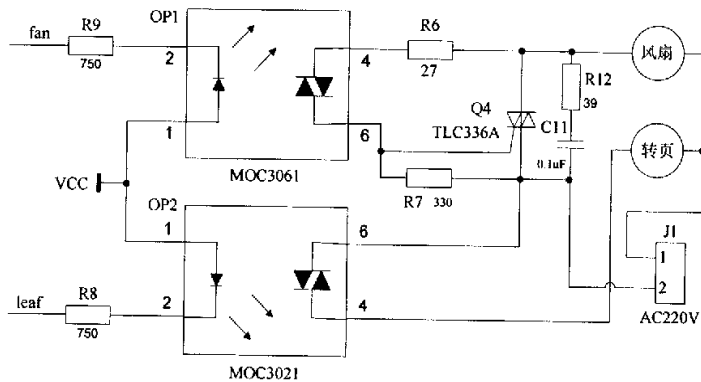


图 5-5 电机控制与驱动电路

5.1.9 软件设计实现

软件设计采用了模块化的设计思想,用 c51 进行编写,系统软件由主程序模块和中断服务子程序组成。从功能上分包括系统初始化,测温程序,按键处理程序,串口通讯程序,电机调速程序和显示程序等模块。显示采用定时动态扫描方式,每隔 2 毫秒刷新一次。由于温度模拟信号变化缓慢,系统每隔 100 毫秒采集一次。定时控制由定时器 T0 来完成,遥控接收由中断服务子程序实现。

5.1.9.1 主程序模块

主程序流程图如图 5-6 所示。系统上电或复位后,首先系统初始化,对中断、定时器和波特率进行设置。然后在主循环中依次完成以下工作:调用测温程序测量室内当前温度;调用按键扫描程序,检查是否有按键按下,若有按键按下,识别按键并设置相应的参数和标志位;调用串口通讯程序,确定是否接收到嵌入式网络接口模块发送过来的命令和数据,若接收到命令和数据,设置相应的参数和标志位,完成相应功能;调用电机调速程序使电机按所设定的参数和标志位工作;调用显示

程序，分时显示当前温度、定时时间和风速。

5.1.9.2 红外中断服务程序模块

红外接收软件使用输入捕捉中断处理红外遥控信号，根据不同的红外编码，可以完成调节风速、风型、时间等功能。

本设计中遥控的发射部分采用现成的遥控器，内部芯片型号为 M50462。M50462 发射的控制信号采用脉冲位置调制 (PPM) 方式，抗干扰能力强。其“0”码是占空比为 1/4 的负脉冲，“1”码是占空比为 1/8 的负脉冲，负脉冲宽度为 0.25ms，其波形图如图 5-7 所示。每一个键号由 17 个负脉冲组成一个 16 比特的二进制数，高 8 位为 IC 的识别码，是为防止不同机型的干扰而设，低 8 位是功能码。按下按键，遥控器以 44ms 的周期按 D15~D0 的顺序重复发送功能码，高位在前，低位在后，直到按键释放为止。

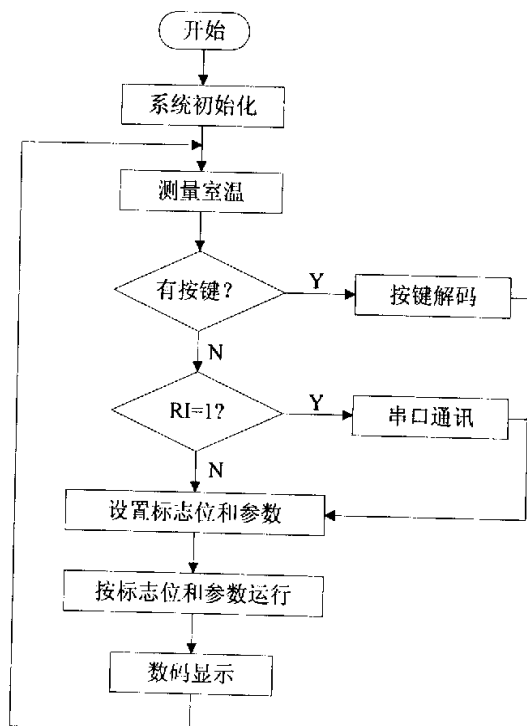


图 5-6 主程序流程图

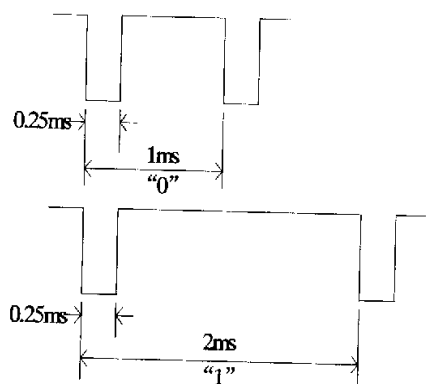


图 5-7 “0”和“1”脉冲波形图

遥控信号的接收有两种方法，一种是定时采样：有定时器设定一个频率，每隔一定时间，定时就产生一个中断，并且对遥控信号输入的 I/O 口进行采样，根据采样所得的高低电平以及相互间的时间间隔来判定输入的信号是“0”还是“1”；第二种是使用定时器上捕获的功能：由定时器上捕获到高低电平的时间，与编码规则

上的时间比较, 来判定输入的信号是“0”还是“1”。很明显, 第二种方法的精度比较高。

本系统采用第二种方法。把中断方式设为脉冲触发, 在脉冲的下降沿产生中断, 无信号时, 红外线接收头的输出脚为高电位, 有信号到来时, 产生一个负脉冲, 进入中断服务程序, 同时启动定时器T1, 在下一个脉冲下降沿到来产生的中断里, 停止T1计时, 将结果取出处理, 马上再次启动定时器T1。如此进行16次后, 接收完一功能码。考虑到电路上晶振的频率误差及不同遥控器的误差, 测得的结果不可能刚好等于理论值, 可设置一个窗口, 在窗口内的结果均为有效。窗口窄误码少, 但遥控距离近, 窗口宽则相反, 因此可经过反复实验, 兼顾误码和遥控距离的原则设置窗口。

红外输入捕捉中断模块的流程图如图5-8所示。

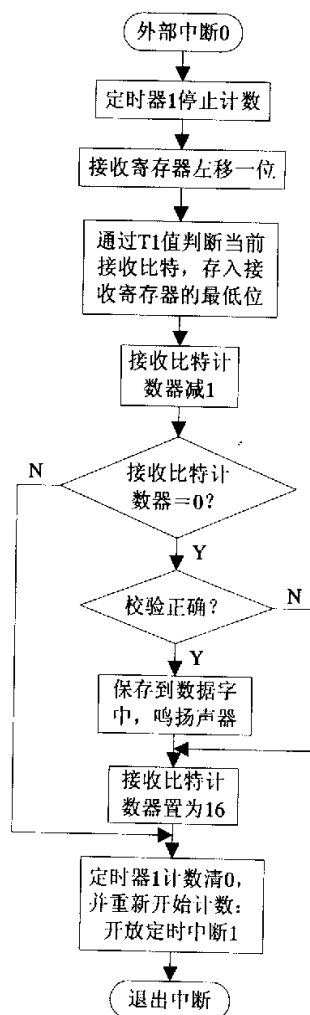


图5-8 红外中断服务程序流程图

5.2 空调控制器的设计

5.2.1 空调的工作原理

空调制冷原理如图5-9所示, 空调工作时, 制冷系统内的低压、低温制冷剂蒸汽被压缩机吸入, 经压缩为高压、高温的过热蒸汽后排至冷凝器; 同时室外侧风扇吸入的室外空气流经冷凝器, 带走制冷剂放出的热量, 使高压、高温的制冷剂蒸汽凝结为高压液体。高压液体经过节流毛细管降压降温流入蒸发器, 并在相应的低压下蒸发, 吸取周围热量; 同时室内侧风扇使室内空气不断进入蒸发器的肋片间进行热交换, 并将放热后的变冷的气体送向室内。如此, 室内外空气不断循环流动, 达到降低温度的目的。

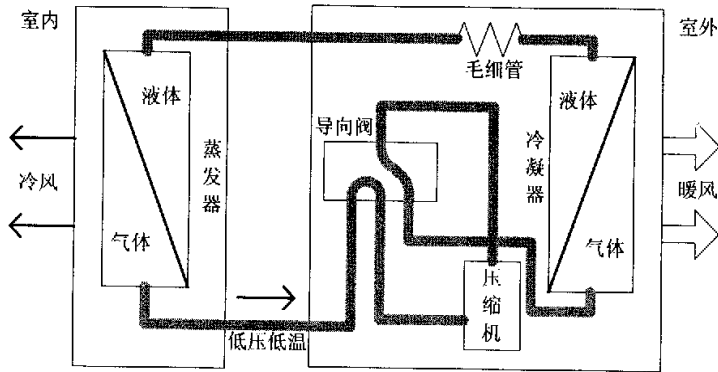
5.2.2 控制器的性能指标要求

设计的空调控制器能实现以下几个部分的功能:

(1) 实现常规功能

空调控制器能够实现制冷、除湿、通风、自动运行模式, 选择制冷功能后, 风机按设定模式运行, 当房间温度大于设定温度时, 压缩机开始运转, 当房间温度低

于设定温度时，压缩机、风机停止运行；选择除湿模式运转，空调器以制冷方式运转到房间温度达到设定温度时，既转入除湿运转，压缩机继续运行，室内风机以低速运转，压缩机及风机按每运转 10min 停止 6min 的规律往复转换；选择通风模式运转，只有风机运转；选择自动模式运转，根据环境温度和系统运行状态自动选择进入制冷、除湿或是通风模式运行。



5-9 空调制冷原理图

(2) 室内温度的测量和控制

系统时刻检测室内温度传感器的检测值，根据室温与设定温度之差及室温的变化来控制压缩机的运转。

(3) 手动设置功能

设计空调控制器的目的是面向用户，通过键盘操作设定参数是人机接口的一个重要方式。键盘部分包括：“开/关”键、功能键、温度设定键、风速设定键和定时键。

(4) 实现数码显示功能

显示功能是人机接口的另一个重要方式。显示功能包括：设定温度显示、当前室温显示、定时时间显示、风速显示、功能显示、定时状态显示等。

设定温度、当前室温和定时时间显示采用一只二位数码管显示，显示范围 15℃~30℃，当按定时键后，显示时间，停止按定时键 3 秒钟后，将轮流显示设定温度、当前室温和定时时间。风速和功能采用发光二极管显示。

(5) 实现红外遥控接收功能

遥控器是空调控制系统的一个重要部分，这是现代家用电器先进性的一个重要标志。设计一个遥控接收模块，主要用于实现红外遥控信号的转换及接收功能。

(6) 网络控制功能

通过串行通信接口与嵌入式网络接口模块进行通讯，使空调接入 Internet。

5.2.3 控制器的硬件组成

空调控制器的硬件原理框图如图 5-10 所示。由图可知，空调控制器的硬件是以 AT89S52 微处理器为核心，主要由电源电路、面板按键、红外接收电路、温度检测电路、串行通信接口、数码显示电路以及继电器驱动电路等几部分组成。

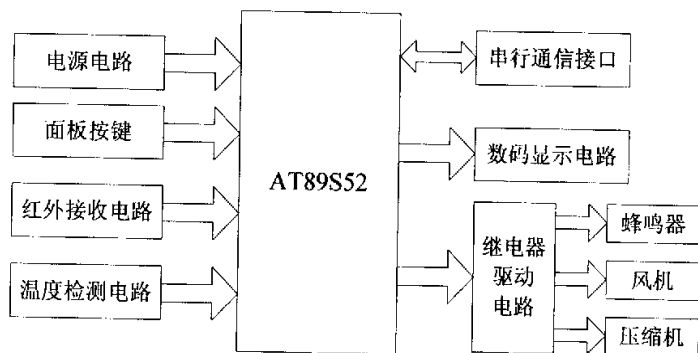
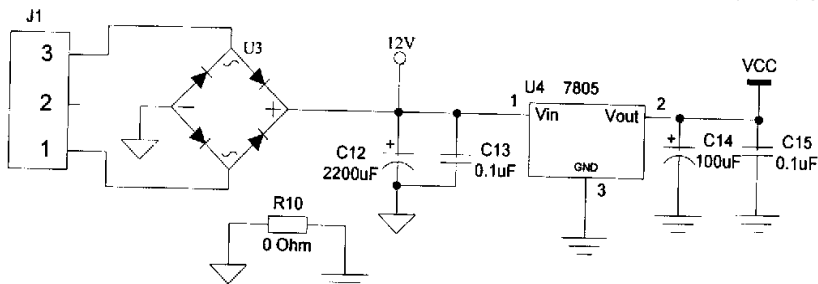


图 5-10 空调控制器原理框图

人机接口模块(面板按键、红外遥控模块)和串行通信接口电路见 5.1.6 和 5.1.7。

5.2.4 电源电路

整个主控板上有三种电压：AC220V、DC12V 和 DC5V。AC220V 直接给压缩机、风机供电；DC12V 和 DC5V 用于继电器和微控系统供电。电源电路如图 5-11 所示，电源变压器将交流电网 220V 的电压变为所需要的电压值，然后从插座 J1 输入，经过整流桥进行全波整流，通过并联 2200 μ F/35V 电解电容、0.1 μ F 电容组成的滤波电路滤除纹波得到 DC12V，再经过三端稳压管 7805 将电压稳压在+5V。



5-11 电源电路

5.2.5 温度检测电路

设计选用了 Dallas Semiconductor 公司推出的数字式温度传感器 DS18B20，温度

测量范围为 $-55^{\circ}\text{C}\sim+125^{\circ}\text{C}$ ，可编程为9位~12位A/D转换精度，测温分辨率可达 0.0625°C ，被测温度用符号扩展的16位数字量方式串行输出；其工作电源既可在远端引入，也可采用寄生电源方式产生；它的方便之处在于单线接口设计，使处理器只需要接一条数据线就能对它进行全部的操作，实现操作指令和测量数据的传输，节省了大量的引线和逻辑电路。以上特点使DS18B20在温度测控系统中得到广泛应用。

DS18B20和单片机典型连接有两种方式：(1)寄生电源方式，其VDD和GND端均接地。(2)外接电源方式，VDD端采用 $3\text{V}\sim 5.5\text{V}$ 电源供电。本设计采用了外接电源方式供电。

5.2.6 数码显示电路

显示模块主要由数码LED显示块和74HC164构成，74HC164为单向8位移位寄存器，可实现串行输入，并行输出。74HC164编程简单，性价比高。选用单片机的两个I/O口完成与显示模块的串行显示连接，其中把单片机的P2.4口作为移位脉冲输出端，P2.5口作为数据输出端。单片机的P2.2口接74HC164的CLK端，对单片机的P2.2口输出的高低电平就是给74HC164数据移位的时钟脉冲，在74HC164获得时钟脉冲的瞬间，如果数据输入端是高电平，则就会有一个1进入到74HC164的内部，如果数据输入端是低电平，则就会有一个0进入其内部，如此循环8次即可将一个8位数据传送至74HC164中。74HC164的输出（并行输出）直接作为数码管的段选控制信号，单片机的P2.0口和P2.1口驱动输出显示的是位选控制信号。显示段码驱动电路如图5-12所示。

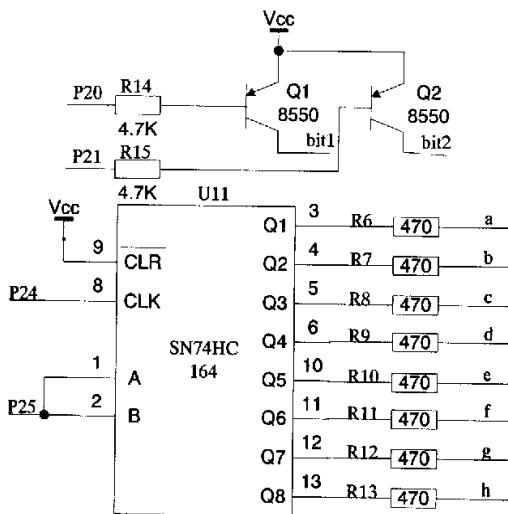


图5-12 显示驱动电路

5.2.7 继电器驱动电路

在单片机应用系统中，开关量输出电路主要完成动作信号的输出，用于控制压缩机和风机的开关状态。由于输出信号不足以驱动继电器的开关动作，在单片机的

信号输出和继电器之间采用了 ULN2003A 实现信号的放大及驱动。ULN2003A 是由 7 组达林顿晶体管阵列和相应的电阻网络以及箝位二极管构成，具有同时驱动 7 组负载的能力，为单片双极型大功率高速集成电路。它具有电流增益高、带负载能力强、温度范围宽及工作电压高等特点，适合驱动继电器、显示器等大功率器件。

继电器驱动电路设计如图 5-13 所示，单片机控制信号经 P1.0~P1.3 端口输出，并通过 P3.4 的控制，将信号锁存在 74LS273 中，74LS273 的输出再经过达林顿驱动器 ULN2003A 反向放大后加到继电器的输入端，使压缩机和风机按要求动作。74LS273 锁存控制信号，一方面增加输出功率，另一方面也防止单片机复位时引起控制的误动作。

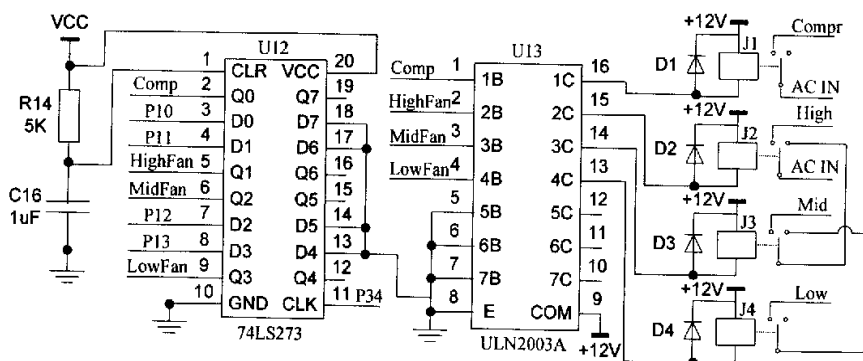


图 5-13 继电器驱动电路

5.2.8 软件设计与实现

系统软件用 c51 编写，采用了模块化的设计思想，由主程序模块、各个功能子程序模块和中断服务子程序模块三大部分组成。主程序的功能是系统初始化、控制程序走向和调用功能子程序；功能子程序包括数据采集、数码显示、风机和压缩机的控制等子程序；中断服务子程序包括遥控接收、定时中断处理等。下面对主要程序模块进行描述。

5.2.8.1 主程序模块

主程序是整个控制系统软件的枢纽，通过主程序有机地调用系统中各类的子程序及模块，使它们形成一个联系紧密的整体，有条不紊地完成各项预定的操作指令。

主程序模块流程如图 5-14 所示。系统上电或复位后，系统首先进行初始化，包括各个寄存器和芯片的初始化；然后对室温进行采样及加权处理；起始串行通信模块确定是否接收到网关发送过来的命令和数据，若接收到命令和数据，设置相应的

参数和标志位；调用键盘扫描程序，检查是否有按键按下，若有按键按下，识别按键并设置相应的参数和标志位；调用显示模块用于显示设定温度、当前室温、定时时间、风速、功能和定时状态等信息；调用功能查询及处理模块使系统按所设定的参数和标志位工作。只要系统上电，主程序就不能停止，一直处于循环等待状态，所以主程序没有结束运行的指令。

(1) 系统初始化

系统初始化主要用于完成各个工作寄存器、各个中断控制寄存器、定时控制寄存器和串行通信端口工作方式等部分的初始化。

(2) 采样及数据处理模块

系统温度的采集使用数字温度传感器 DS18B20，DS18B20 与单片机之间采用串行数据传送，因此，在对 DS18B20 进行读写编程时，必须严格遵守 DS18B20 的操作协议和操作时序，否则将无法读取测温结果。实现这个时序有两个方法：一是可以用软件延时，二是定时延时。其中软件延时简单实用，效果好一些，不过要求晶体振荡器要好，才能保证延时的准确性。

根据 DS18B20 的通信协议，单片机控制 DS18B20 完成温度转换必须经过三个步骤：每一次读写之前都要对 DS18B20 进行复位，复位成功后发送一条 ROM 指令，最后发送 RAM 指令，这样才能对 DS18B20 进行预定的操作。当单片机收到 DS18B20 的响应信号后，便可以发出操作命令，这些命令可以分为 ROM 命令和 RAM 命令两种。单片机的操作过程如图 5-15 所示。

在温度测量中，对 DS18B20 的操作主要是复位、读数据和写数据三种，而后两种操作又是按位进行的，所以首先应该按照 DS18B20 的时序要求，编写读、写时间片的程序，其流程如图 5-16 所示。

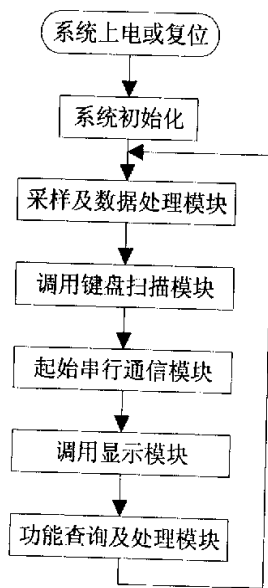


图5-14 主程序流程图

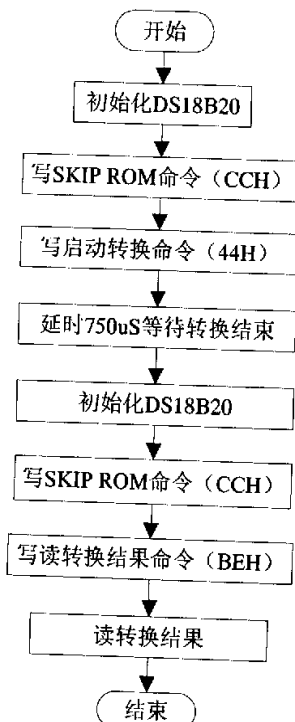


图5-15 测温流程图

由于系统在工作时要把室内温度设定值与实际采样值进行比较,为了保证控制过程的有效性,采样值应尽可能贴近真实值。为此采用连续采样方式,并对采样数据采用加权平均求值法进行处理,以滤除测量信号中的干扰成分,保证采样精度和控制效果。加权平均求值公式如下所示:

$$y_n = \frac{1}{2}x_n + \frac{1}{4}x_{n-1} + \frac{1}{8}x_{n-2} + \frac{1}{16}x_{n-3} + \frac{1}{16}y_{n-1} \quad (5-1)$$

其中 y_n 为本次采样的确定值; x_n 为本次采样的实际值; $x_{n-1} \sim x_{n-3}$ 为本次采样相邻的前三次采样值; y_{n-1} 为上次的采样确定值。

有公式可知,为了得到本次采样的确定值,至少需要知道连续4个实际采样值,因此系统上电状态下确定采样值需要连续采样至少四次。第一次进行加权平均处理时,为了使 y_{n-1} 不为零,令此时对应的 $y_{n-1} = x_{n-3}$,则以上公式可写为 5-2 式:

$$y_n = \frac{1}{2}x_n + \frac{1}{4}x_{n-1} + \frac{1}{8}x_{n-2} + \frac{1}{16}x_{n-3} + \frac{1}{16}x_{n-3} \quad (5-2)$$

这种采样数据处理方式考虑了相近几次采样值的影响,因此可以将外界干扰引起的偶然误差及其它原因导致的采样结果突变对采样确定值的影响限制到最小,使采样值尽可能接近真实值。

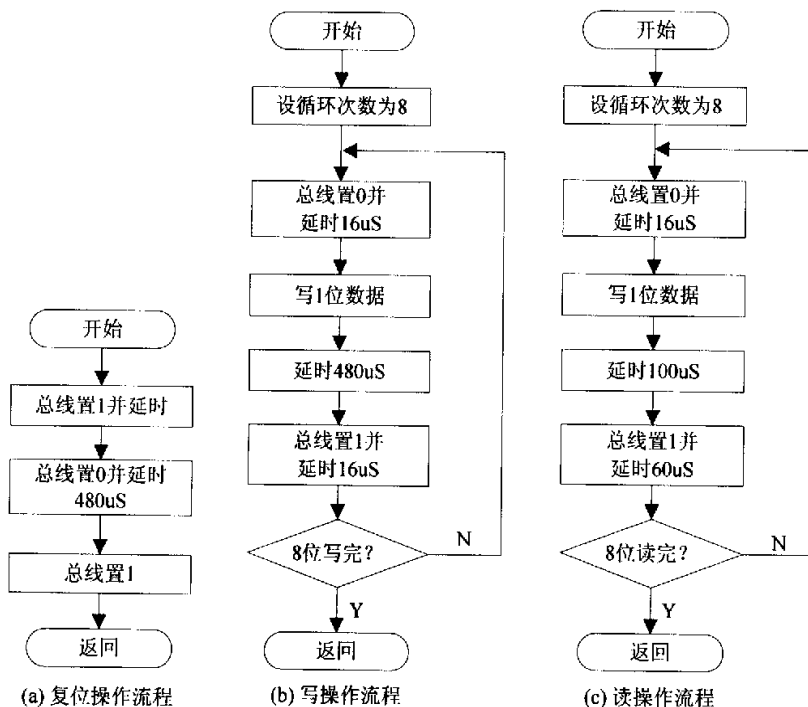


图 5-16 DS18B20 操作程序流程图

(2) 键盘扫描模块

主程序中的键盘扫描模块主要用于上电复位情况下检查 ON/OFF 键是否按下, 如按下则执行显示模块显示初始状态, 否则数码管无显示。周期性键盘扫描设置在主程序中, 以满足及时识别按键、迅速响应的要求。

键盘是人机对话的接口, 对它的编程主要有两个要求: ①键盘响应必须要及时, 不能让使用者产生迟钝的感觉; ②键盘处理程序不能占用 CPU 太多的时间。否则影响其他功能的实时性。考虑到按键的数目较少, 所以将键盘的相应处理放在了主程序中。程序流程如图 5-17 所示, 功能键采用循环控制方式完成制冷、除湿、通风和自动四个功能的设定; 风速键采用循环控制方式完成高风速、中风速和低风速三个功能的设定。这种设计方式使按键的数目减少了, 按键的减少使键盘布置更加美观, 操作更加方便、直观。

键盘处理主要是由按键识别和按键处理两部分构成。按键识别由按键查询模块完成, 当检测到键盘口有低电平时, 经过按键去抖动后确定哪个按键被按下了; 按键处理是根据按键识别结果进行相应的参数和标志位设定, 如定时器计数、温度设定值加一计算等。

(4) 串行数据通信模块

空调控制器与嵌入式网络接口模块通过 RS232 进行串行通信, 单片机的串行端口有 4 种工作方式, 通过编程设计串行端口的 2 个控制寄存器, 设置工作在方式 1 下, 单片机和嵌入式网络接口模块设置相同的波特率 9600b/s, 由定时器 T2 的定时溢出时间来确定。

单片机采用循环方式实现串行数据的传送, 在循环方式下, 单片机循环对数据寄存器 SBUF 进行读写来实现数据的接收和发送。在每次嵌入式网络接口和控制器通信时, 若发出的是读状态命令, 则控制器将自己的状态数据上传; 若发出的是写指令, 则设置相应的参数和标志位, 并把参数回传。

(5) 显示模块

显示模块包括数码管显示部分和 LED 发光二极管显示两部分, 软件设计采用定时刷新方式。在主循环中调用显示模块, 进行数码管显示。

两位数码管的显示范围为 $12^{\circ}\text{C}\sim 30^{\circ}\text{C}$, 初始状态下温度为 24°C 。温度和定时时间共用一个数码显示模块, 当按定时键后, 显示时间, 停止按定时键 3 秒钟后,

将恢复温度显示。数码管显示程序流程如图 5-18 所示，将显示数据转换成段码送到段控制端口，选通该位控制端口，并延时一段时间。

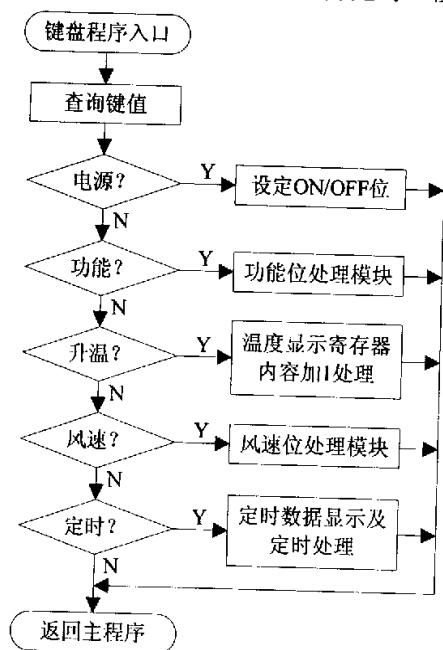


图 5-17 键盘扫描程序流程图

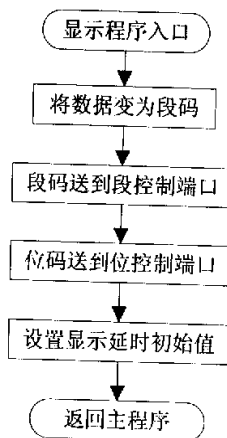


图 5-18 显示程序流程图

(6) 功能查询及处理模块

空调控制器控制功能的实现，如：制冷、除湿及风速调节等，主要是通过循环查询参数和标志位确定要执行的功能，并执行相应的驱动程序控制压缩机和风机的运转。程序模块流程如图 5-19 所示。

5.2.8.2 中断服务程序模块

(1) 红外接收中断程序

红外接收软件使用输入捕捉中断处理红外遥控信号，根据不同的红外编码，可以完成开关机、调节运行模式、风速、时间等功能。接收程序流程图见 5-8 所示。

(2) 定时中断程序

控制系统中涉及的时间参数比较多，我们通过定时器 0 来实现所有的定时控制，中断周期为半秒。在中断服务程序中计算空调定时时间、压缩机的运行时间、关闭时间、以及延时保护时间等，并设置相应的标志位，工作流程如图 5-20 所示。

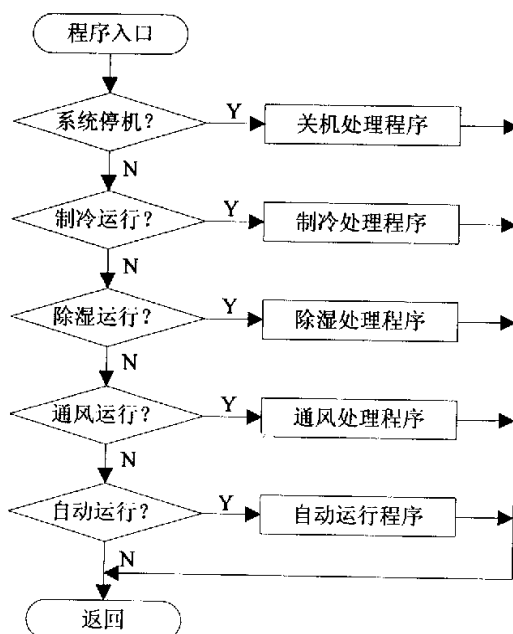


图 5-19 功能查询及处理模块流程图

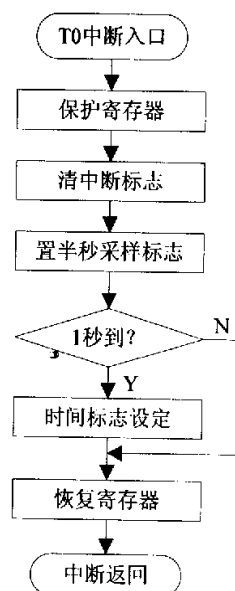


图 5-20 定时中断流程图

5.3 抗干扰措施

控制系统在使用时，经常受到温度、潮湿及电磁噪声等干扰因素的侵害，但整个系统的结构又要求简单轻便，这就要求单片机应用系统既有较强的抗干扰能力，且使用的硬件资源又要尽量少。所以在设计上，应采取必要的软硬件抗干扰措施，减少或免除各种不良因素对系统的影响和损害，提高系统工作的可靠性和稳定性。

5.3.1 硬件抗干扰措施

硬件抗干扰是系统抗干扰的很重要的一个方面，是系统设计过程中始终要考虑的问题之一。

印刷电路板是单片机应用系统中器件、信号线、电源线的高密度集合体，印刷电路板设计的好坏对抗干扰能力影响很大，故印刷电路板的设计绝不单是器件、线路的简单布局安排，还必须符合抗干扰的设计原则。本设计采用了下述抗干扰措施：

(1) 在元器件布局方面，把相互有关的元件尽量放的靠近一些，如晶振，CPU 时钟输入端都易产生噪声，在放置的时候应把它们靠近些。对于那些易产生噪声的器件，小电流电路，大电流电路，开关电路等，应尽量使其远离单片机的逻辑控制电路。对于继电器控制电路另外制成电路板，这样有利于抗干扰，提高电路工作的可靠性。

(2) 尽量在关键元件旁边安装去藕电容。

在印刷电路板的电源线 VCC 和地线之间接入去藕电容，原则上每个集成电路芯片的旁边都放置一个 0.01 μ F 的瓷片电容，并且电容引线不要太长。

(3) 接地是抑制干扰的重要方法，如能将接地和屏蔽正确结合起来使用可解决大部分干扰问题。设计中，主要采取了以下方法：①数字电路和模拟电路分开，在制作电路板时，将模拟电路用地线与数字电路隔离，以增强抗干扰能力。②接地线尽量加粗，若接地线很细，则地线电阻将会很大，造成接地电位随电流的变化而变化，致使信号电平不稳，导致电路抗干扰能力下降。③接地线构成闭环回路。对于只用数字电路组成的印刷电路板接地时，将其做成闭环回路大多明显地提高抗噪声能力。其原因是：一块印刷电路板上有很多集成电路，尤其遇有耗电多的元件时，因受到线条粗细限制，地线产生电位差，引起抗噪声能力下降，如连成环路，则其差值缩小。因此，将系统的主板(高速逻辑电路板)的地线设计成回路。

(4) 电路板的布线设计优劣，对系统的抗干扰性能影响也很大，合理的布线是力求将系统中各元件之间，电路之间可能产生的不利影响限制在最低程度。从抗干扰的角度考虑，设计时元件排列及信号走线尽量有序，短直，简洁，避免相邻电路互相影响；尽量避免过长的平行走线，减少布线的分布电容；避免印刷线路形成环路，以免环路接受噪声形成干扰。

(5) CMOS 芯片的输入阻抗很高，易受感应，对其不用的引脚要接地或接电源；继电器、按钮等在操作时会产生火花，必须利用 RC 电路加以吸收，继电器线圈两端还应并接续流二极管。

(6) 元器件的质量对系统影响很大。应选择正品元器件。使用前还要进行必要的筛选。对于接插件，应选择抗震性能好，接合可靠，防松的接插件。传输电缆应具有性能良好的屏蔽层，耐老化，抗损伤，不易断线。

5.3.2 软件抗干扰措施

如果仅靠硬件来解决系统的抗干扰问题，势必造成硬件电路的过分庞大，不但增加成本，而且会引起新的抗干扰问题。另外，应用系统在现场使用时，大量的干扰源虽不能造成硬件系统的损坏，但经常使系统不能正常工作，致使控制失灵。系统的抗干扰不可能单纯依靠硬件来解决，因此有必要进行软件抗干扰设计。本文本

着在满足系统控制功能的前提下，尽量提高软件的可靠性的原则，采取了以下软件抗干扰措施。

(1) 单片机在接收到第一个输入信号时，并不承认它为有效信号，而以一定的时间间隔连续读入3次同样有效的输入信号才确定该信号为有效信号。采用这种多读入法避免了干扰输入。

(2) 对于输出的开关量控制信号，要多次重复输出同一数据。只要有可能，其重复周期尽可能短些。外部设备接受到一个被干扰的错误信息后还没有作出反应，一个明确的输出信息又来到，就可以及时防止错误动作的发生。

设置当前输出状态寄存单元，当干扰侵入输出通道造成输出状态破坏时，系统能及时查询寄存单元的输出状态信息，及时纠正输出状态。

(3) 设置软件陷阱，在未使用的大片程序存储器区用020000填满，当弹飞的程序落到非程序区时，使弹飞的程序掉入软件陷阱，强行将程序纳入正轨。

(4) 增加程序运行监视系统(Watchdog)，当程序弹飞到一个临时构成的死循环时，使用Watchdog能及时发觉死循环并能使系统复位。

5.4 系统测试

本文设计的系统在局域网和广域网中都进行了测试，主要用来测试对家电的远程监控。用户和家电控制系统的交互界面以网页的形式实现，在客户端浏览器地址栏里输入家电的IP地址并提交连接请求，嵌入式网络接口模块就会对浏览器的请求做出响应，将存放在Flash中的网页发送给浏览器。用户点击网页中的超链接部分，如当点击空调控制，嵌入式网络接口模块就会将对空调控制的网页发送给浏览器。

系统在局域网和广域网中都通过了测试，达到了预期的效果。登录系统时，系统响应时间一般小于2s；对家电的控制准确率为100%；系统在局域网内运行比较稳定，但在广域网中运行时，在网络不通畅的情况下会出现数据丢失现象。

5.5 本章小结

本章详细地描述了电风扇控制器和空调控制器的软硬件设计，并根据实际情况采取了相应的抗干扰措施。最后对家电控制系统进行了测试，达到了预期效果。

第6章 结束语

本文在对嵌入式 Internet 技术的发展及接入方式进行了概括分析之后, 根据家电控制系统的特点选择了一种基于 SST89E554RC 微处理器的嵌入式 Internet 解决方案, 并详细介绍了系统的软硬件设计。本系统将家电与广泛分布的以太网结合起来, 实现了对家电控制的网络化。

本文完成的研究工作主要有:

(1) 对嵌入式 Internet 技术进行了理论分析和方案论证, 根据家电控制系统的具体特征, 提出了在 8 位微处理器上实现嵌入式 Internet 的家电控制系统解决方案。

(2) 进行芯片的比较选型和电路设计, 在充分理解微控制器及网络芯片后, 对系统整体设计及硬件原理进行了具体的设计分析, 并利用 Protel99SE 进行原理图和 PCB 的绘制, 完成印刷电路板制作调试。系统的设计, 除了实现基本功能外, 还具有扩展能力, 便于功能的进一步完善。

(3) 从设计的角度出发, 详细分析并精简了 TCP/IP 协议栈, 以及如何在 8 位处理器上实现这些协议, 具体的协议有以太网协议、ARP、IP、UDP、TCP、HTTP 等。

(4) 以电风扇和空调为例, 实现了这两种家电控制器的软硬件设计。家电控制器不仅能完成常规功能, 而且能通过通讯接口按照通讯协议与嵌入式网络接口模块进行通信, 实现用户对家电的远程监控。

基于本文方案构建的嵌入式 Internet 平台使用外围器件少, 系统成本低, 易于开发和维护, 不需要经过网关代理, 总体价格低廉。经过测试, 该系统能够正确的监控家电, 网络延迟也可以满足家电设备需求, 取得了较为满意的运行效果。可广泛应用于家电控制、电力自动化及各种电器测控设备、通讯终端、在线检测系统、智能仪器仪表等众多领域。

本文工作还有一些不足之处, 有待进一步的开发和完善。总结一下, 主要有以下几个方面:

(1) 本设计采用定时器中断, 作为系统各任务之间相互协调的机制, 即采用定时器人工实现分时系统; 如果采用嵌入式操作系统, 来取代微控制器的分时处理方式, 系统整体性能将会得到很大的提高。

(2) 无线通讯产品将成为嵌入式软件的重要应用领域，一方面，已有无线产品将借助芯片技术和嵌入式软件来提高性能；另一方面当前许多嵌入式产品都将增加无线通讯功能。因此，很有必要使产品具有无线通讯功能，以嵌入式网络接口模块为核心，组建家庭无线网络，从而更好地为用户服务。

(3) 作为利用在家电控制中的嵌入式系统，在网络/安全方面的需求是必不可少的。需要增加安全验证模块，提高系统安全性。

(4) 系统的抗干扰性、稳定性、抗风险能力也有待加强，如在强劲网络风暴环境下经受考验的能力尚待检验等。此外在工艺及成本上与商品化、产品化要求尚有一定差距，系统理念及全局思想尚待提高。

总之在该领域尚有许多工作等待我们在日后的工作和学习中探索完善！

致谢

首先我要向导师王忠勇教授表示深深的敬意和真诚的感谢!他在承担繁重的教学、科研工作的同时,给我极大的鼓励、支持和帮助,从论文选题、方案设计、资料收集,直至论文的编写,都是在他的悉心指导下进行的。在攻读硕士学位期间,我在学习和生活上时时刻刻受到导师的悉心指导和亲切关怀,我所取得的点滴进步,无不凝聚着导师的大量心血。导师渊博的知识,严谨的治学态度,使我终生难忘!良师益友是我对导师最中肯的评价。

感谢宋豫全老师对我的教诲,在整个研究生阶段,宋老师不仅在学习上给予过我细心的指导,而且在生活上给了我无私的帮助。在此,我向宋老师表示衷心的感谢和敬意!

感谢程明教授、邓记才副教授,在我学习期间和他们有过深入的接触,同时得到了他们的指导和帮助,受益匪浅。向他们表示真心的感谢。也向学院其他所有老师致以衷心的感谢和敬意!

感谢和我一起度过研究生生活的同学,张延宾、李志军、范运兴、高原、史晓鹏、路新华、董红政等同学。感谢他们给予我的热情帮助和支持。

最后,感谢我的家人从物质上、精神上给予我的支持和鼓励,使我得以顺利完成学业。在我的学习和工作中,他们永远是我最好的精神支柱。

参考文献

- [1]桑楠, 嵌入式系统原理及应用开发技术, 航空航天大学出版社, 2002
- [2]张大波, 嵌入式系统原理设计与应用, 机械工业出版社, 2005
- [3]Hongtao Yang, Eagleson R. Design and Implementation of an Internet-based embedded control system. Control Applications, 2003
- [4]Robert F. Embedded Internet systems come home. [J]. IEEE Internet Computing, 2001, 40(14): 52-53
- [5]Szymanski, J. W., Embedded Internet technology in process control devices, Factory Communication Systems; 2000. Proceedings. 2000 IEEE International Workshop on, 6-8 Sept. 2000: Pages: 301-308
- [6]王勇, 嵌入式Internet的技术实现及其安全问题的研究, 浙江大学博士论文, 2002
- [7]周志洪, 基于嵌入式网络技术的网络化自动抄表系统的研究, 浙江大学博士论文, 2005
- [8]吕京建, 嵌入式以太网技术的兴起与前景, 今日电子, 2000
- [9]肖婕、毛宇斌, 嵌入式系统关键技术分析与开发应用, 单片机与嵌入式系统应用, 2003年第7期
- [10]张运杰, 基于8位微控制器的嵌入式Internet研究与实现及其在远程视频监控中的应用, 天津师范大学硕士论文, 2004
- [11]窦振中、宋鹏、李凯, 嵌入式系统设计的新发展及挑战, 单片机与嵌入式系统应用, 2004年第12期
- [12]杨四海、马利, 信息家电嵌入式系统选择方案, 自动化技术与应用, 2002年第4期
- [13]蒋巧文、孟祥贵, 基于单片机的嵌入式系统网络接入方案, 电子产品世界, 2004年第7期
- [14]李阳明、齐志强、师丽彩, 嵌入式Internet的应用, 电子技术, 2005年第5期
- [15]唐仕斌、曾文海, 嵌入式Internet以太网接口设计及实现, 低压电器, 2003年第2期

- [16]刘发贵、张晖, 信息家电中的Web浏览器, 华南理工大学学报, 2004年第6期
- [17]梁磊、洪永强, 基于网络技术的嵌入式系统, 机电技术, 2004年第10期
- [18]刘炽、向鹏飞, 家电网络技术的应用, 家电科技, 2004年第7期
- [19]万柳、蔡斌, 基于嵌入式Internet技术的智能家庭局域网系统的实现, 信息工程大学学报, 2004年第9期
- [20]李三波, 基于因特网的信息家电系统, 电脑开发与应用, 2003年第9期
- [21]Andrew S. Tanenbaum, 计算机网络(第3版), 清华大学出版社, 1998
- [22]Pete Loshin, TCP/IP透彻理解(第4版), 电子工业出版社, 2004
- [23]谢希仁译, TCP/IP协议族, 清华大学出版社, 2001
- [24]范建华等译, TCP/IP详解卷一: 协议, 机械工业出版社, 2000
- [25]刘恩涛、王沁, 单片机中TCP/IP协议子集的设计与实现, 计算机工程与设计, 2004年第12期
- [26]魏忠等编著, 嵌入式开发详解, 电子工业出版社, 2003
- [27]Hunang Jiao, Liu-Jianguo, GaoMin, Embedded Temperature Web Controller Based on Internet, Thee 2004 IEEE International Symposium on Industrial Electronics, 2004
- [28]<http://www.laogu.com>
- [29]Silicon Storage Technology, Inc. FlashFlex51 MCU SST89E554RC Preliminary Specifications, 2001
- [30]Silicon Storage Technology, Inc. Boot-Strap Loader Software Example User's Guide, 2002
- [31]Realtek semi-conductor CO. LTD. Realtek Full-Duplex Ethernet Controller with Plug and Play Function (RealPNP) Specification, 2001
- [32]求是科技, 单片机典型模块设计实例导航, 人民邮电出版社, 2004
- [33]杨金岩、郑应强, 8051单片机数据传输接口扩展技术与应用实例, 人民邮电出版社, 2005
- [34]赵亮, 单片机C语言编程与实例, 人民邮电出版社, 2003
- [35]李宇, uVision2单片机应用程序开发指南, 科学出版社, 2005
- [36]王伦, 电风扇原理与维修技术[M], 新时代出版社, 1999
- [37]申桂英, 史庆军, 张颖, 风扇控制器的微机控制, 工业仪表与自动化装置, 2000

年第12期

- [38] 马凤英、朱小红, 风扇控制系统的智能化设计, 山东轻工业学院学报, 2005年第3期
- [39] 丁建军、陈定方、周国柱, 基于AT89C51的智能电风扇控制系统, 湖北工学院学报, 2003年第4期
- [40] 李学龙, 使用单片机控制的智能遥控电风扇控制器, 电子制作, 2003年第9期
- [41] 肖凤明, 绿色空调器单片机控制电路原理与维修图说, 电子工业出版社, 2004
- [42] 金建轩、陈建设, 网络空调解决方案, 电子产品世界, 2002年第5期
- [43] 吴兆垠, 网络空调器的设计, 家电科技, 2004年第1期
- [44] 金刚、陈建民、苑广礼等, 空调远程监控系统, 家电科技, 2004年第1期
- [45] 秦晓红、翟正军, 利用单片机实现空调的自动控制, 现代电子技术, 2000年第9期
- [46] 郑飞霞, 家用空调器的抗干扰, 家电科技, 2004年第7期

附录 攻读硕士学位期间论文发表、项目研究情况

论文发表情况:

1. 第一作者, 网络家电系统的嵌入式接口设计, 微计算机应用, 已录用, 2007.4
2. 第二作者, 汽车行驶记录仪的设计, 周口师范学院学报, 2005.5

项目研究情况:

基于 IPv6 的信息家电系统设计, 主要参与人