

特别说明

此资料来自豆丁网(<http://www.docin.com/>)

您现在所看到的文档是使用**下载器**所生成的文档

此文档的原件位于

<http://www.docin.com/p-55214017.html>

感谢您的支持

抱米花

<http://blog.sina.com.cn/lotusbaob>

大功率电器智能识别与用电安全控制器的设计

摘 要: 针对宿舍用电管理及用电安全存在的种种问题, 设计了以单片机为核心能自动识别大功率电器并能限制其使用并能组成网络管理的智能用电控制器。介绍了系统的组成、主要模块的工作原理、上位机软件设计及程序流程图等。

关键词: 功率识别、用电安全、串行 A/D

Designment of High-Power Electric Appliances Intelligently Identifying and Electrical Safety Controller

Abstract: Aimed for problems of electrical control and safety widely existed in dormitories, we designed a chip microcomputer that can form network management based intelligent electricity controller. In this paper, construction of this device, theory of its main part, designment of the server computer's software, flows of the programmer are introduced.

Keywords: Power intelligent、Uses electrical safety、Serial A/D

第 1 章 绪论

1.1 课题来源及主要研究内容

1.1.1 课题来源

本课题是针对一般公寓用电管理及用电安全存在的一些问题而提出的。设计一个以单片机为核心的智能用电控制器。能在设定的总功率范围内智能识别大功率电器并限制其使用，并能通过串行通信和网络管理实现对各居住楼以及生活区的管理。

1.1.2 主要研究内容

本课题运用到单片机原理与接口技术、电子线路 CAD、模拟电子技术和数字电子技术、电气控制技术、程序设计等。本课题对公寓用电主回路电流的检测和主回路上负载的智能识别等关键技术进行研究，并研制一种大功率电器智能识别与用电安全控制器。控制器采用单片机进行数据处理和控制，应用抗干扰滤波技术，能巧妙地避开电路的各种冲击和干扰信号，准确地判断电路各负载的功率特性，从而智能识别大功率电器，并限制其使用，有效地防止了大功率电器的使用，节约用电，减少安全隐患，但又不影响常规电器的正常使用。

主要完成的内容如下：

- a. 确定控制器以及网络管理的总体设计方案
- b. 硬件原理设计，完成参数计算与元器件选择
- c. 控制器 PCB 图设计
- d. 控制器硬件安装制作与调试
- e. 应用软件的设计与调试
- f. 联网管理软件及操作界面的设计

1.2 课题研究的意义

市场上各种廉价或质量不合格的电器产品越来越多，由电器故障造成的触电、电起火等事故屡见不鲜。现在电能越来越紧张，社会各界都越来越重视安全用电及节约用电。

学生公寓是人口密集、用电负载类型多样的场所。当使用以上大功率的电器设备时，很容易引起火灾等事故，直接威胁同学们的人身安全和学院的财产，并且给学校带来负面的社会影响。因此广大大专院校，一般都采取禁止学生在宿舍内使用电炉、热得快、电热杯之类的大功率电器的措施。但禁用的方法只是定期对学生宿舍进行搜查，发现就没收、

罚款之类的。这其实并不是个有效、合理的方法。正所谓“上有政策，下有对策”，浪费人力去搜查，搜查人员可能徒劳无功的居多。

在学生宿舍内，由于是集体生活，节约用电意识较弱，这样就难免造成电能的一些不必要浪费。如宿舍停电后，很多小电器没及时关掉，下次来电时，这些小电器的功率和就不小了，人如果不在宿舍的话，就浪费了电能。全国这么多学校、这么多宿舍，浪费的电能加起来也不是个小数目。

为了解决目前学生宿舍用电管理难，电能浪费大，学生使用大功率电器安全隐患多等问题。设计了大功率电器智能识别与用电安全控制系统。

供电与用电之间的矛盾日益突出，传统的用电管理系统远远不能满足高校后勤管理数字化的要求，建立集群式供电智能管理系统，是电量管理数字化的迫切需要。大功率电器智能识别与用电安全控制系统是集群式供电智能管理系统，是用电智能管理发展的趋势，具有很大的市场和研究意义。

1.3 课题发展现状和前景展望

随着电子技术、计算机网络技术和通讯技术的发展，人们已研制出供配电智能控制系统。其中有两方面非常突出：

(1) 集群式供电智能管理系统的下位机软件编程。下位机的软件程序包括与上位机串口通信、单片机数据采集和数据保存等工作。集群式供电智能管理系统以上位机(PC)为主机，下位机(单片机)为从机，利用 RS485 总线通过串口进行通讯。

(2) 对用电过程中的负载类型进行识别。一般情况下，电压波形为正弦波，即标准电压；当一些学生公寓由于年代久远，供电设备和线路老化，线路容量不足时，会使电压波形产生畸变，成为非正弦波，即削顶电压。因此控制系统要实现在以上两种电压波形下负载类型的识别。其目的就是限制大功率性负载(如电炉子、热水器等)。但允许使用计算机负载，可以根据要求设定限制允许使用的计算机功率的值(一般设定为 300-500W)，当检测到用户使用非法电器时，将立即断电，一段时间后，恢复供电，如继续使用，将再次断电，并记录。

第2章 系统及其实现网络管理的总体设计方案

2.1 单片机应用系统的设计概述

单片机具有集成度高、功能强、可靠性高、体积小、功耗低、使用方便、价格低廉等特点，在各个领域得到了广泛的应用和发展。目前已渗透到人们工作和生活的各个角落，几乎是无处不在。单片机最早是以嵌入式微控制器(embedded microcontroller)的面貌出现的。在嵌入式系统中，它是应用最多的核心器件。在计算机主导工业生产并且日益走进家庭生活的今天，从家用电器、工业控制、医疗仪器到军事应用，到处都有单片机的存在。

目前，MCS-51 系列单片机以其独特优越的性能，在智能仪表、工业测控、数据采集、计算机通讯等领域得到极为广泛的应用。智能仪器在人们心目中的概念是，凡是内部含有单片机的仪器统称为智能仪器。故也有人把智能仪器称为微机化仪器。智能化仪器内部基本上都是用单片机进行信息控制与处理。特别是近年来在市场上出现的数字信号处理器 DSP 是一种速度极高的单片机，它在通信和高速信息处理中起了极大的作用，从而扩展了单片机在智能仪器中的应用。目前，无论在高、中、低档仪器中，还是在一些常规仪器和特种仪器中都大量应用单片机。以单片机为核心组成智能仪表已是一种必然的结构型式。

单片机应用系统是指以单片机为核心，配以一定的外围电路和软件，能实现某些功能的应用系统。它由硬件部分和软件部分组成。因此，单片机应用系统的设计应包括硬件设计和软件设计两大部分。为保证系统能可靠工作，在软、硬件的设计中，还要考虑系统的抗干扰能力，即设计过程中还包括系统的抗干扰设计。

一般说来，随着用途不同，应用系统的硬件和软件结构也不同，但是研制过程方法步骤基本上相同。其研制过程的步骤如下：

- 1) 分析题目的设计要求，并进行可行性调研与分析论证，确定设计方案。
- 2) 应用系统总体方案设计
- 3) 系统硬件、软件和抗干扰设计
- 4) 系统的硬件与软件调试
- 5) 印刷电路板及外形设计
- 6) 系统组装与调试
- 7) 固化应用程序，试运行。

2.2 单片机与 PC 机通信接口技术

通信是两个设备之间的数据交换，串行通信方式特点是通信线路只要一对传输线就可以实现通信，从而大大地降低了成本，特别适用于远距离通信。

2.2.1 串行通信接口标准

在串行数据通信中，发送端发出的是一系列二进制脉冲，而接收端必须能够正确地识别出每个字符和字符内的每一位。串行通信双方为了有效地交换信息必须建立一些约定，即对数据的编码、同步方式、传输速度、传输控制步骤、校验方式、报文方式等问题给予统一的规定，这些规定称为通信协议。

异步串行通信常用的波特率为 50, 75, 110, 150, 300, 600, 1200, 2400, 4800, 9600 等。通信过程中，接收方按约定的格式接收数据，并进行检查，一般可以查出以下三种错误，并会给出相应的出错信息，提示用户处理。

- 1) 奇偶错：在约定奇偶检查的情况下，接收到的字符奇偶状态和约定不符。
- 2) 帧格式错：一个字符从起始位到停止位的总位数不对。
- 3) 溢出错：若先接收的字符尚未被微机读取，后面的字符又传送过来，则产生溢出错。

通用的串行 I/O 接口标准有许多种，最常见的是 RS-232 和 RS-485，二者各有其应用领域，使用的范畴也不尽相同，现就这两种标准作简单介绍。

(一) EIA RS—232C 通信

这是美国电子工业协会推荐的一种标准(Electronic Industries Association Recommended Standard)。它在一种 25 针接插件(DB-25)上定义了串行通信的有关信号。新一代的计算机均以 9 引脚的接头(DB—9)接出所有的 RS-232 通信端口，它们在计算机和连接线上的位置和定义如图 2-1 所示。

- 1 脚：CD—Carrier Detect，载波检查
- 2 脚：RXD--Receive，数据接收。
- 3 脚：TXD--Transmit，数据传输。
- 4 脚：DTR-Data Terminal Ready 数据端待命。
- 5 脚：GND--Ground，地线。
- 6 脚：DSR-Data Set Ready 传输端待命。
- 7 脚：RTS-Request To Send，要求传输。
- 8 脚：CTS—Clear To Send，清除并传输。
- 9 脚：RI-Ring Indicator，响铃指示。

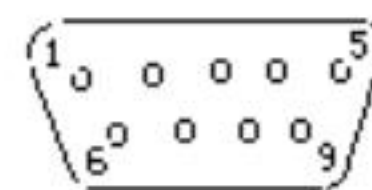


图 2-1 RS-232 9 引脚的接头

在实际异步串行通信中，并不要求用全部的 RS—232 信号，一般通信线有三条(TXD、

RXD 和信号地)就能工作了。

RS—232C 规定了双极性的信号逻辑电平。其中：-3V 到-15V 之间的电平表示逻辑“1”；+3V 到+15V 之间的电平表示逻辑“0”。因此这是一套负逻辑定义。

以上标准称为 EIA 电平。PC/XT 系列使用的信号电平是-12V 和+12V，符合 EIA 标准，但在计算机内部信号是 TTL 电平，因此这中间需要用电平转换电路。常用芯片 MCI488 或 SN75150 将 TTL 电子转换为 EIA 电平，MCI489 或 SN75154 将 EIA 电平转换为 TTL 电平。PC/XT 系列以这种方式进行串行通信时，在波特率不高于 9600 的情况下，理论上通信线的长度限制约为 15 米。

(二) RS—485 通信

由于串行通信的简单易用，在工业上也使用了串行通信作为数据交换的手段；可是工业环境通常会有噪声干扰传输线路，在用 RS-232 作传输时经常会受到外界的电气干扰而使信号发生错误。

RS—485 采用差动式工作驱动器 (G) 与接收器 (R)，无论在抗干扰、传输距离、传输速率方面都较 RS-232 有很大提高。RS—485 性能的提高是基于以下二方面的改进：首先，将信号的幅值（峰对峰值）从 RS-232 的 25~30V 降低为 12V；其次，RS—485 为发送回路及接收回路分别提供独立的地线，而不再共用一条公共地线，两项措施有力地减少了线间干扰，从而也提高了传输距离、传输速率。

RS-485 的信号传输方式如图 2-2 所示。RS-485 的信号在传送出去之前会先分解成正负的两条线路，当到达接收端后，再将信号相减还原成原来的信号。如果将原始的信号标注为(DT)，而被分解后的信号分别标注为(D+)和(D-)，则原始信号与分解后的信号在由传输端传送出去时的运算关系如下：

$$(DT) = (D+) - (D-) \quad \text{式①}$$

同样地，接收端在接收到信号后，也按上式的关系将信号还原成原来的样子。而如果此线路受到干扰，其情况可能如图 2-3 所示。

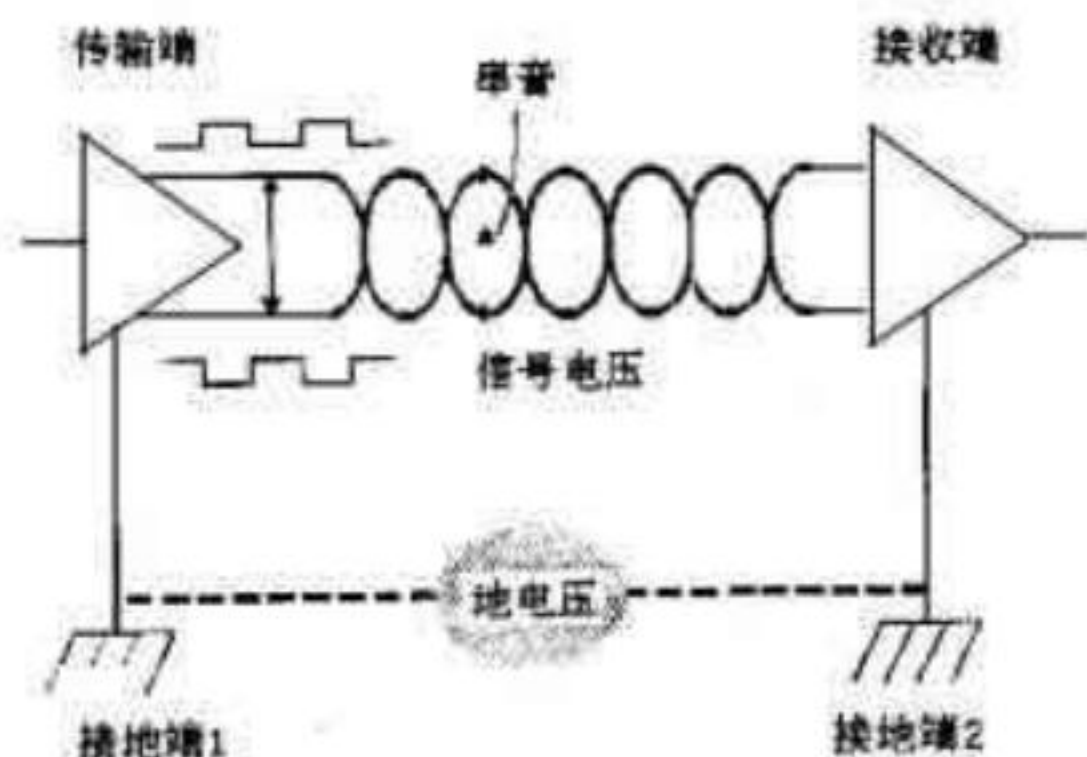


图 2-2 RS—485 信号传输方式

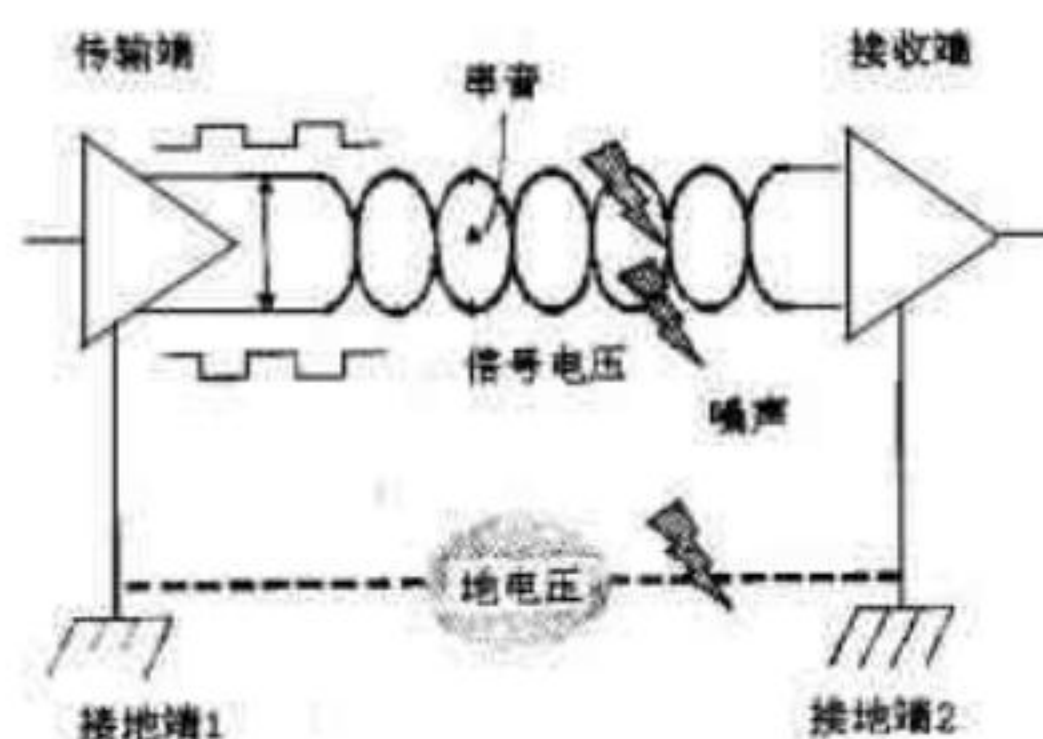


图 2-3 RS—485 的信号与噪声

这时候在两条传输线上的信号会分别成为 $(D+)+\text{Noise}$ 和 $(D-)+\text{Noise}$ ，如果接收端接收此信号，它必须按照一定的方式将其合成，合成的方程式如下：

$$(DT)=[(D+)+\text{Noise}]-[(D-)+\text{Noise}]= (D+)-(D-) \quad \text{式②}$$

式①式②的结果是一样的。所以使用 RS-485 网络可以有效地防止噪声干扰，也正因为这种特性，工业上比较适合使用这种串行传输方式。

2.2.2 选择通讯方式

选用 RS-485 网络，信号传输抗干扰性强些。为了增大传输距离，可在串行接口电路与外部设备之间增加信号转换电路。目前常用的转换电路 RS-485 收发器。。RS-485 收发器将微型计算机的 TTL 电平转换为差分信号进行传送，最大通信距离为 1.2km(在 100Kb/s 传输速率以下)。采用 MAX485 芯片完成电平转换。

图 2-4 是 MAX485 引脚和工作电路，MAX485 包含 RS-485 发送器 1 个，接收器 1 个。1 个发送器可驱动 32 个节点，抗静电保护电压 15kV。图 2-5 是典型半双工 RS-485 网络，平衡连接电缆两端有终端电阻。

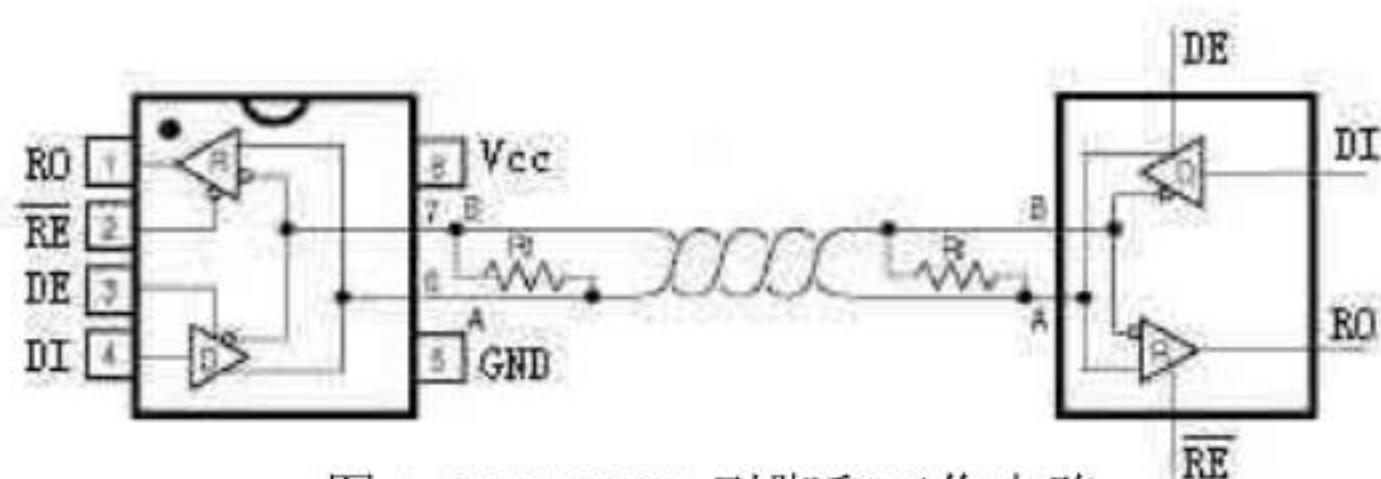


图 2-4 MAX485 引脚和工作电路

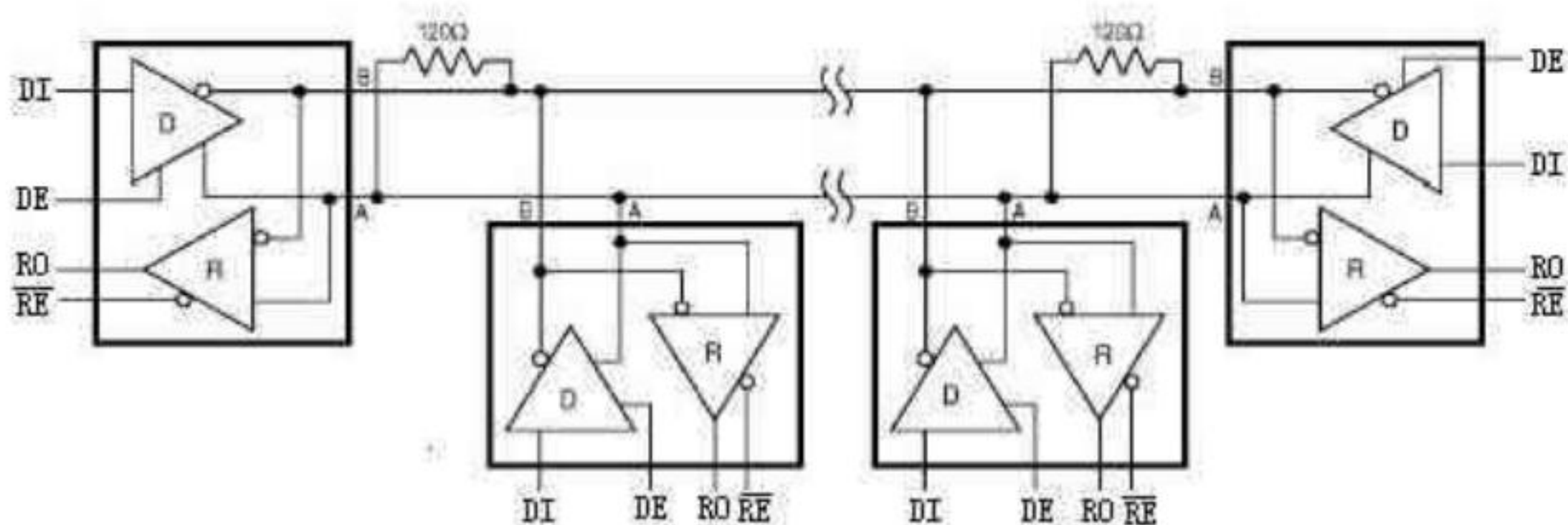


图 2-5 典型半双工 RS-485 网络

2.3 系统网络管理的总体方案设计

整个系统由上位计算机、中继器模块和终端控制器组成。①上位机观察各个公寓内的用电情况并可控制终端控制器动作。②中继器模块完成电平的转换和终端控制器的地址分配。整个系统由一台计算机和多台单片机构成 RS485 总线等多机通信结构,单片机在与计算机串行通信时,因各自串行标准不同,需在中继模块进行电平转换,将 RS485 标准转换成 RS232 标准。③终端控制器采用单片机进行数据处理和控制,应用抗干扰滤波技术,能巧妙地避开电路的各种冲击和干扰信号,准确地判断电路各负载的功率特性,从而智能识别大功率电器,并限制其使用,有效地防止了大功率电器的使用,节约用电,减少安全隐患,但又不影响常规电器的正常使用。

系统网络管理的总体设计方案如图 2-6 所示。

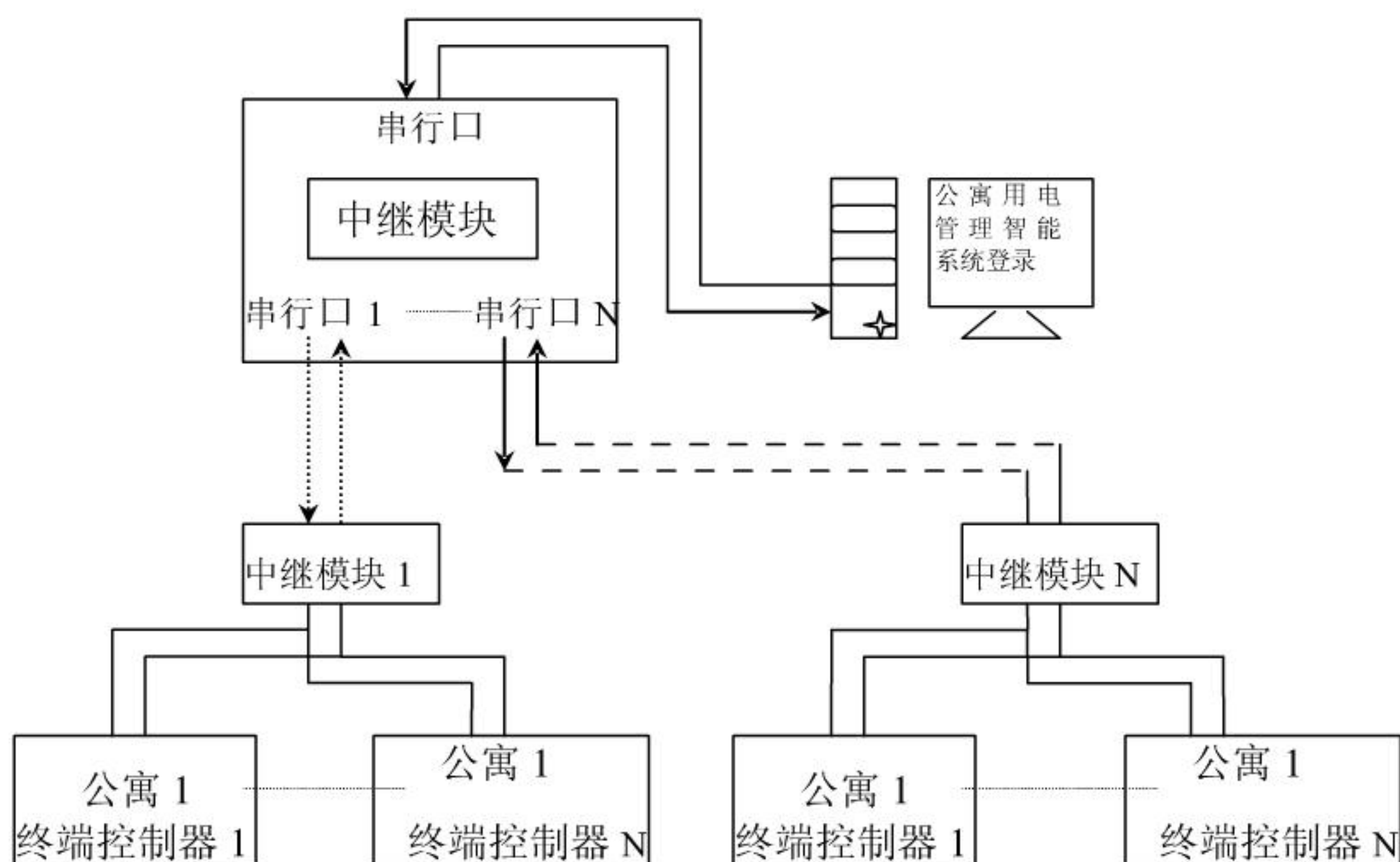


图 2-6 系统网络管理的总体设计方案

第3章 硬件设计及控制器 PCB 图设计

3.1 硬件设计方法

一个单片机应用系统的硬件设计包括两大部分内容：一是单片机系统扩展部分的设计，它包括存储器扩展和 I/O 接口扩展。二是各功能模块的设计，如信号测量功能模块、信号控制功能模块、人机对话功能模块、通讯功能模块等，根据系统功能要求配置相应的 A/D、D/A 接口，键盘、显示器、打印机等外围设备。

在进行硬件的总体方案设计时，所涉及到的具体电路可借鉴他人在这方面进行的工作，因为经过别人调试和考验过的电路往往具有一定的合理性。在此基础上，结合自己的设计题目进行一些修改，则是一种简便、快捷的做法。当然，在完全照搬不可能的情况下，有些电路还需要自己设计。系统的硬件电路设计应注意以下几个方面：

- 1) 尽可能选择标准化、模块化的典型电路，提高设计的成功率和结构的灵活性。
- 2) 在条件允许的情况下，尽可能选用功能强、集成度高的电路或芯片。因为采用这种器件可能代替某一部分电路，不仅元器件数量减少，而且接插件和相互连线减少，可以使系统可靠性增强，而且成本往往比用多个元器件实现的电路要低。
- 3) 在对硬件系统总体结构考虑时，同样要注意通用性的问题。对于一个较复杂的系统，设计者往往希望将其模块化，然后采用一定的连接方式将其组合成一个完整的系统。在这种情况下，连接方式就显得非常重要，有时可选用通用接口方式，不少厂家已开发出适合于这些总线结构的接口板，如输入板、输出板、A/D 板等。在必要的情况下，选用现成的模块板作为系统的一部分，尽管成本有些偏高，但会大大缩短研制周期，提高工作效率。当然，在有些特殊情况和小系统的场合，用户须自行设计接口，定义连线方式。此时要注意接口协议，一旦接口方式确定下来，各个模块的设计都应遵守该接口方式。
- 4) 在电路设计时，要充分考虑应用系统各部分的驱动能力。一些经验欠缺者往往忽视电路的驱动能力及时序问题，认为原理上通就行了，其实不然。因为不同的电路有不同的驱动能力，对后一级系统的输入阻抗要求也不一样。如果阻抗匹配不当，系统驱动能力不够，将导致系统工作不可靠甚至无法工作。因此，在电路设计时，要注意增加系统的驱动能力和减少系统的功耗。
- 5.) 设计时应尽可能地采用最新的技术。因为电子技术发展迅速，器件更新换代很快，市场上不断推出性能更优、功能更强的芯片，只有时刻注意这方面的发展动态，采用新技术、新工艺，才能使产品具有最先进的性能，不落后于时代发展的潮流。

6.) 注意选择通用性强、市场货源充足的元器件，尤其对需大批量生产的场合，更应注意这方面的问题。其优点是：一旦某种元器件无法获得，也能用其它元器件直接替换或对电路稍做改动后用其它器件代替

7) 工艺设计也是十分重要的问题，包括机箱、面板、印刷电路板、配线、接插件等，这也是一个初次进行系统设计人员容易疏忽的问题。在设计时要充分考虑到安装、调试和维修的方便。

3.2 终端系统的组成及工作原理

3.2.1 硬件设计思路

利用各个负载的功率特性不同，利用电流互感器采集主电路电流的线性变化量，通过单片机处理采样数据并控制继电器。设置禁用下限值，对学生宿舍内使用的电器进行智能化的区别，即能自动识别所用的是常规电器还是禁用的大功率电器。从而限制大功率禁用电器的使用。它与普通限流装置有着本质的区别，它并不只是简单的对线路负荷电流进行限制，而是一种智能控制系统。

学生在宿舍内可以正常使用多数的灯、风扇、电脑、打印机之类的常规电器，而当有像电炉、热得快等纯电阻大功率用电器使用时装置就能利用负载突增时的功率跳变自动识别，并立即跳闸断电，延时 15 秒后再重合闸恢复送电。设置手动重合闸和限定次数的自动重合闸，来实现断电恢复和断电保护功能。设置总功率上限值为 1200W—1800W。而且还可以配合其他装置来设置触电、过载、短路等保护功能，更加有效的防范了一些用电安全事故的发生。

3.2.2 基本构成

该系统主要有主控制模块、信息检测模块、A/D 转换模块、输出控制模块、保护模块等组成，如图 3-1 所示。其中主控制模块完成数据处理和输出控制功能；信息检测模块完成对负载用电回路电流的采集功能；A/D 转换模块主要完成对主电路的电流进行采样；输出控制模块实现控制负载用电回路通断的功能；保护模块完成短路保护功能。

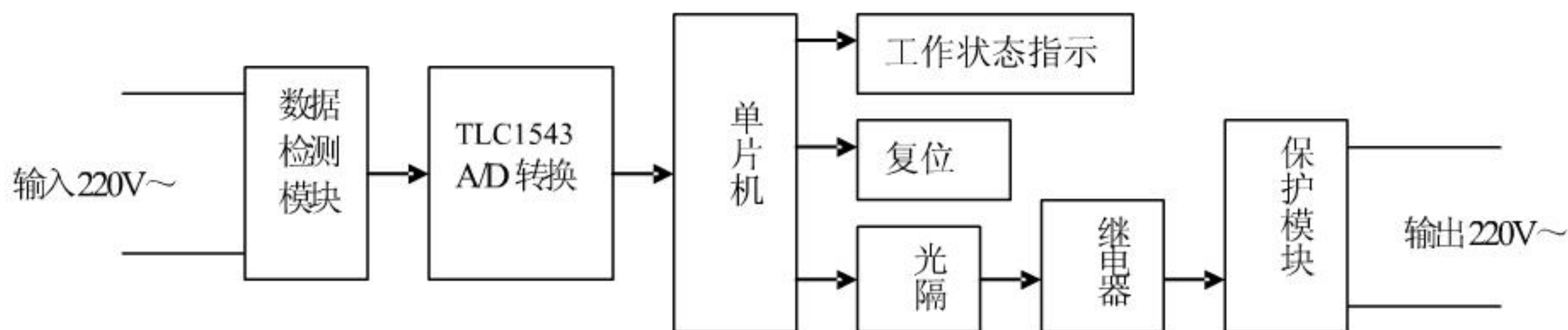


图 3-1 控制系统原理图

3.2.3 基本工作原理

用电流互感器采集主电路中的电流信号，经过信号调理、串行 A/D 芯片 TLC1543 进行 A/D 转换，将采样值送入单片机对其进行处理。根据功率最小禁用电器接入时，设定单个电器的上限值（常规电器一般单个功率都不大于 300W）。当有常规小功率电器接入时，电流的瞬间变化量将不会超过设定值，电路能正常供电。而当有单个功率在设定值以上的电器接入时，瞬时电流的变化量将超过允许使用单个电器的上限值，单片机主控器将记忆该值。为了防止误动作，在规定的时间内连续采样 N 次，通过进一步的比较和分判断：若为某一电器（小于设定值）接入时的冲击或电网干扰，输出模块不动作，继续供电；若是恒定的大功率电器接入，则单片机输出控制信号使继电器动作并停止供电。延时后再自动重合闸恢复送电。并设置了手动复位和的自动重合闸次数，来实现断电恢复和断电保护功能。通过设置总功率上限值，一方面可限制总用电功率，另一方面可实现用电安全保护。该控制器还可以配合其他控制器来设置触电、过载、短路等保护功能，更加有效的防范了一些用电安全事故的发生。

3.3 终端硬件设计

3.3.1 确定控制器

Intel 公司的 MCS-51 系列单片机，是目前世界上用量最大的几种单片机之一。由于 Intel 公司在嵌入式应用方面将重点放在 186、386、奔腾等与 PC 机类兼容的高档芯片的开发上，而渐渐放弃了微控制器的生产，以 MCS-51 技术核心为主导的微控制器技术已被 ATMEL、PHILIPS、三星、华邦等公司所继承，并且在原有基础上又进行了新的开发，从而产生了和 MCS-51 兼容而功能更加强劲的微控制器系列。这些公司都在保持与 8051 单片机兼容的基础上，改善了 8051 许多特性(如时序特性)，如提高了速度、降低了时钟频率、放宽了电源电压的动态范围及降低了产品价格。

3.3.1.1 单片机的选择

根据初步设计方案的分析，可以选择带有 EPROM 的单片机，应用程序直接存储在片内，不用在外部扩展程序存储器，电路可以简化。Intel 公司的 8051 和 8751 芯片均可以选用。其他一些 MCS-51 系列相兼容的芯片，例如 ATMEL 公司生产的 AT89C××系列单片机，如表 3-1 所示。AT89C××系列与 MCS-51 系列单片机相比有两大优势：第一，片内程序存储器采用闪速存储器，使程序的写入更加方便；第二，提供了更小尺寸的芯片(AT89C2051 / 1051)，使整个硬件电路的体积更小。它以较小的体积、良好的性能价格比倍受青睐。本课题我们选用 89C2051 单片机。该芯片的功能与 MCS-51 系列单片机完全兼容，并且还

具有程序加密等功能，物美价廉，经济实用。

表 3-1 ATMEL 公司 3 种型号单片机

芯片型号	生产厂家	EPROM	与 MCS-51 系列的区别
AT89C51	ATMEL	4KB	兼容
AT89C1051	ATMEL	1KB	兼容（20 脚）
AT89C2051	ATMEL	2KB	兼容（20 脚）

3.3.1.2 AT89C2051 单片机简介

AT89C2051 是 ATMEL 公司生产的带 2K 字节可编程闪速存储器的 8 位 CMOS 单片计算机。工作电压范围为 2.7V~6V。全静态工作频率为 0HZ~24MHZ。芯片采用 DIP-20 封装形式，与 8051 进行对比后可发现，AT89C2051 减少了两个对外端口（即 P0、P2 口），因而芯片尺寸有所减小。AT89C2051 内部结构与 MCS-51 系列芯片基本相同。引脚配置如图 3-2 所示。其同名引脚与 8051 相同，不同之处说明如下：

P1 口的 8 位双向 I/O 引脚，P1 口引脚写入“1”后，可用作输入。引脚 P1.2~P1.7 提供内部上拉电阻，当作为输入并被外部下拉为低电平时，它们将输出电流（ I_{IL} ）。P1.0 和 P1.1 需要外接上拉电阻，可用作片内精确模拟比较器的正向输入（AIN0）和反向输入（AIN1）；在闪速编程和编程校验期间，P1 口也可接收编码数据。P1 口输出缓冲器能接收 20 mA 电流，并能直接驱动 LED 显示器。

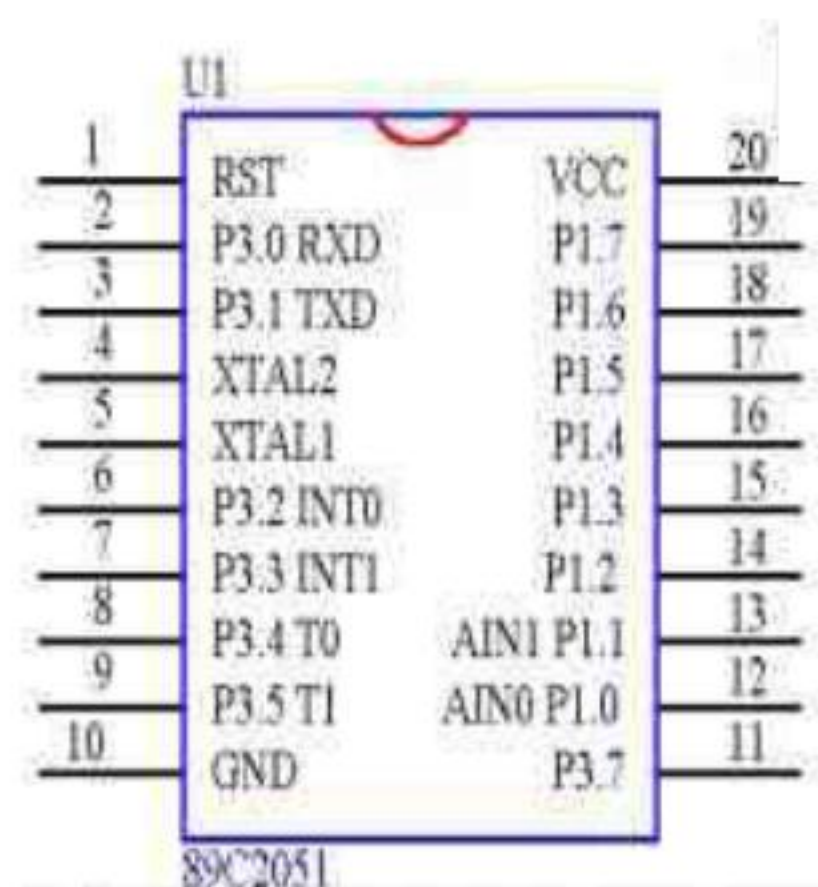


图 3-2 AT89C2051 引脚配置

P3 口也可用作特殊功能口，P3.0~P3.5 引脚功能与 8051 相同，P3.7 作为普通 I/O 引脚。与 8051 比较，没有了 RD、WR 引脚。P3 口引脚 P3.0~P3.5 与 P3.7 为 7 个带内部上拉的双向 I/O 引脚。P3.6 在内部已与片内比较器输出相连，不能作为通用 I/O 引脚访问。P3 口同时也可作为闪速存储器编程和编程校验接收控制信号。P1 口是 8 位双向 I/O 口。

从上述引脚说明可看出，AT89C2051 没有提供外部扩展存储器所需的地址锁存和读写控制信号，因此，利用 AT89C2051 构成的单片机应用系统时，不能像 AT89C51 那样扩展存储器等设备，可以根据情况与 I²C 或 SPI 串行总线设备接口。

同时 AT89C2051 的时钟频率可以为零，即具备可用软件设置的睡眠省电功能，系统的唤醒方式有 RAM、定时/计数器、串行口和外中断口，系统唤醒后即进入继续工作状态。省电模式中，片内 RAM 将被冻结，时钟停止振荡，所有功能停止工作，直至系统被硬件复位方可继续运行。

其性能概括如下：

- 8 位 CPU；
- 工作电压范围 2.7—6V；
- 全静态工作方式：0HZ—24MHZ；
- 一个可编程串行接口；
- 有片内精密模拟比较器；
- 2K 字节的 FLASH 存储器；
- 128 字节的数据存储器；
- 15 根输入/输出线；
- 2 个 16 位定时/计数器；
- 5 个中断源，2 个优先级。

3.3.2 时钟与复位电路的设计

单片机工作的时间基准是由时钟电路提供的。在单片机的 XTAL1 和 XTAL2 两个引脚，接一只晶振及两只电容就构成了单片机的时钟电路。如图 3-3(a)所示。

电路中，电容器 C1 和 C2 对振荡频率有微调作用，通常的取值范围为 $30\text{pf}\pm 10\text{pf}$ ；石英晶体选择 6MHZ 或 12MHZ 都可以。其结果只是机器周期时间不同，影响计数器的计数初值。

单片机的 RST 引脚为主机提供一个外部复位信号输入端口。复位信号是高电平有效，高电平有效的持续时间应为 2 个机器周期以上。

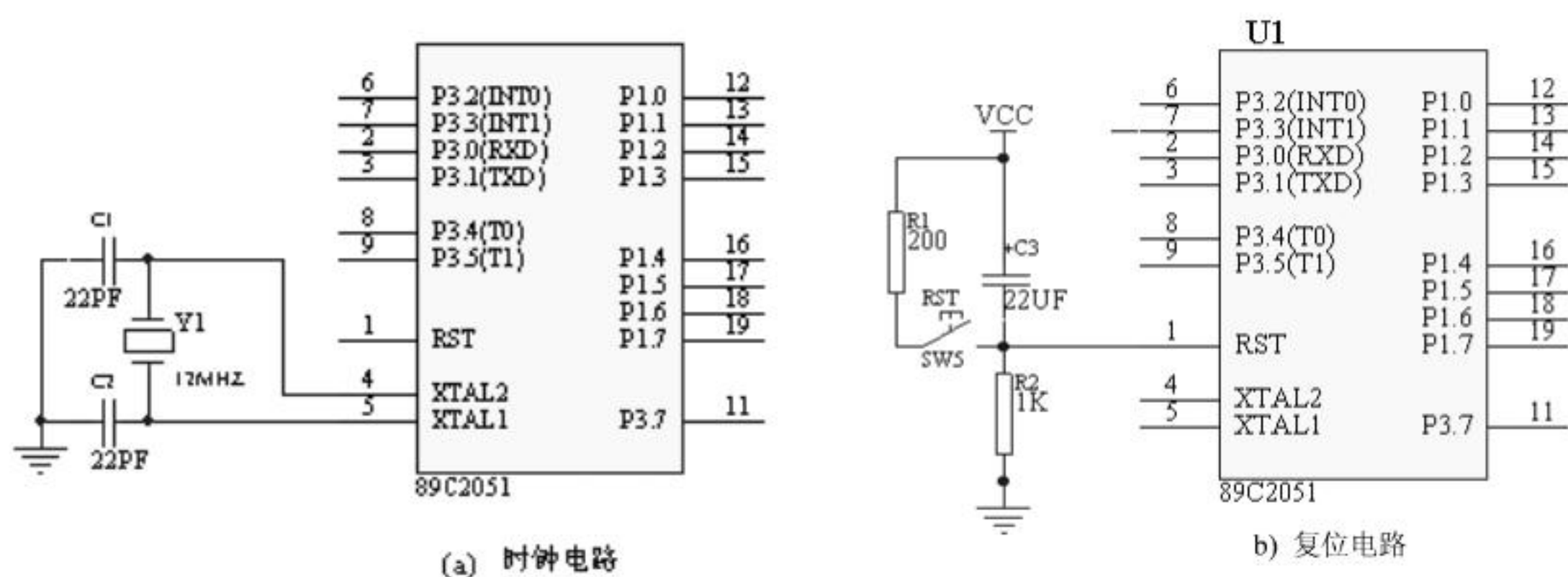


图 3-3 时钟与复位电路

单片机的复位方式有上电自动复位和手工复位两种。图 8(b)是 51 系列单片机常用的上电复位和手工复位组合电路，只要 Vcc 上升时间不超过 1ms，它们都能很好地工作。复位以后，单片机内各部件恢复到初始状态，参见表 3-1。

阻容器件的参考值如图中所示，即 $R1=200\Omega$ ， $R2=1K\Omega$ ， $C3=22\mu F$ 。RET 按键可以选择专门的复位按键，也可以选择轻触开关。

表 3-1 复位后片内各专用寄存器的状态			
寄存器	内容	寄存器	内 容
PC	00H	TMOD	00H
A	00H	TCON	00H
B	00H	TH0	00H
PSW	00H	TL0	00H
SP	07H	TH1	00H
DPTR	0000H	TL1	00H
P0~P3	0FFH	SCON	00H
IP	(XXX00000) B	SBUF	不变
IE	(0XX00000) B	PCON	(0XXXXXXX) B

3.3.3 模/数转换电路设计

3.3.3.1 数/模与模/数转换电路概述

将微型计算机应用于实时控制、在线动态测量等系统时，其控制或测量的对象往往是一些连续变化的模拟量，如温度、压力、流量、位移、速度以及连续变化的电量。当计算机与外设直接交流有关物理量方面的信息时，通常需要将检测的模拟量信号转换成数字信号交计算机进行处理，而计算机输出的数字信号又需转换成模拟量信号以便对执行机构进行控制。实现数字量转换为模拟信号转换的电路称为D/A转换器（DAC），实现模拟量转换为数字信号转换的电路称为A/D转换器（ADC）。

ADC和DAC在测控系统中应用广泛，图9所示为某一以单片机为核心的实时控制系统结构框图，由图可见ADC和DAC在控制系统中的作用与地位。数/模(D/A)和模/数(A/D)转换是一种专门的接口技术，它包括接口电路的硬件设计和实现转换的应用程序的设计。本章主要讨论单片机应用系统中数/模和模/数转换接口电路设计的一般方法和有关问题。

3.3.3.2 A/D 转换器的主要技术参数

A/D 转换器的功能是将模拟量转换为与其大小成正比的数字量信号。能实现这种转换的原理和方法很多，因此 A/D 转换芯片也很多，在选用的时候主要考虑的是 A/D 芯片的技术性能（如分辨率、转换速度）、连接特性等。在设计接口电路时与 D/A 接口类似，也应选择合适的转换芯片，采用合理的电路结构，以满足应用系统的技术性能和使用要求。

由于 A/D 转换器的电路类型不同，编码方法不同，其指标类型也有所区别。如输出数

据为 8421 编码的 BCD 码 A/D 芯片和自然二进制数编码的 A/D 芯片，它们的转换精度含义和表示方法也不同。现以输出二进制数编码的 A/D 转换器为例介绍其主要技术指标。

1. 分辨率 分辨率是 A/D 转换器对微小输入量变化敏感程度的描述。既 A/D 转换器最低有效位 (LSB) 所对应的输入模拟值，对一个分辨率为 n 位的转换器，能够分辨满量程的 2^{-n} 。例如，分辨率为 10 位的 A/D 转换器，输入模拟量的满量程为 5V，则它能分辨的最小电压为 $5000/1024=4.882\text{mV}$ 。

2. 转换时间 指 A/D 转换器完成一次转换所需的时间，即从转换启动信号开始到转换结束并得到稳定的数字输出量所需的时间。一般称转换时间 $<20\mu\text{s}$ 为高速 A/D，转换时间在 $20\sim300\mu\text{s}$ 之间为中速，转换时间 $>300\mu\text{s}$ 为低速。并且还有一些转换速度在 ns 级的超高速的 A/D 转换芯片。

3. 量程 指 A/D 芯片所能转换的模拟输入电压范围，分单极性、双极性两种类型。

3.3.3.3 串行 A/D 转换器 TLC1543 介绍

随着大规模集成电路技术的发展，各种高精度、低成本、低功耗、可编程的 A/D 芯片不断推出，使得仪器仪表和微机测控系统的电路设计更加简洁，可靠性更高。由于单片机独特的 I/O 操作和控制功能，高精度、接口简单而又价格低廉的串行 A/D 芯片在单片机应用系统的受到欢迎。

本课题采用 TLC1534 串行 A/D 转换器，TLC1543 是美国 TI 公司生产的众多串行 A/D 转换器的一种，它具有输入通道多、转换精度高、传输速度快使用、灵活和价格低廉等优点，是一种高性价比的 A/D 转换器。它有输入通道多、转换精度高、传输速度快使用、灵活和价格低廉等优点，是一种高性价比的 A/D 转换器。

它有三个输入端和一个 3 态输出端：片选 (CS)、输入/输出时钟 (I/O CLOKCK)、地址输入 (ADDRESS) 和数据输出 (DATAOUT)。这样通过一个直接的四线接口与主处理器或其外围的串行口通讯。片内含有 14 通道，多路选择器可以选择 11 个输入中的任何一个或 3 个内部自测试 (self—test) 电压中的一个。片内设有自动采样-保持电路。在转换结束时，“转换结束”信号 (EOC) 输出端变高以指示转换的完成。系统时钟由片内产生并由 I/O CLOCK 同步。片内转换器设计使器件有高速 ($10\mu\text{s}$ 转换时间)、高精度 (10 位分辨率、最大+LSB 线性误差) 和低噪声特点。

其工作时序如图 3-4 所示。

timing diagrams

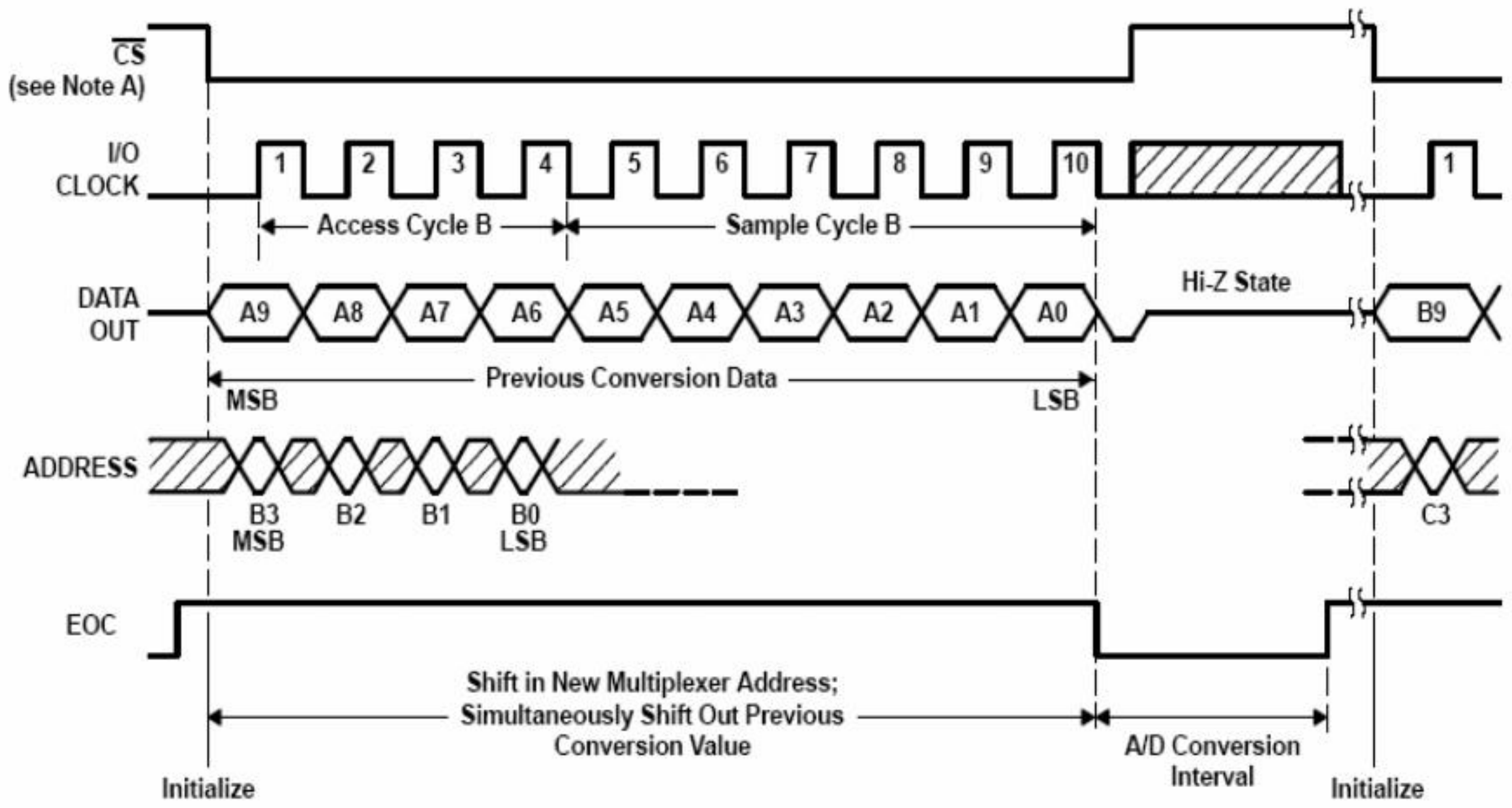


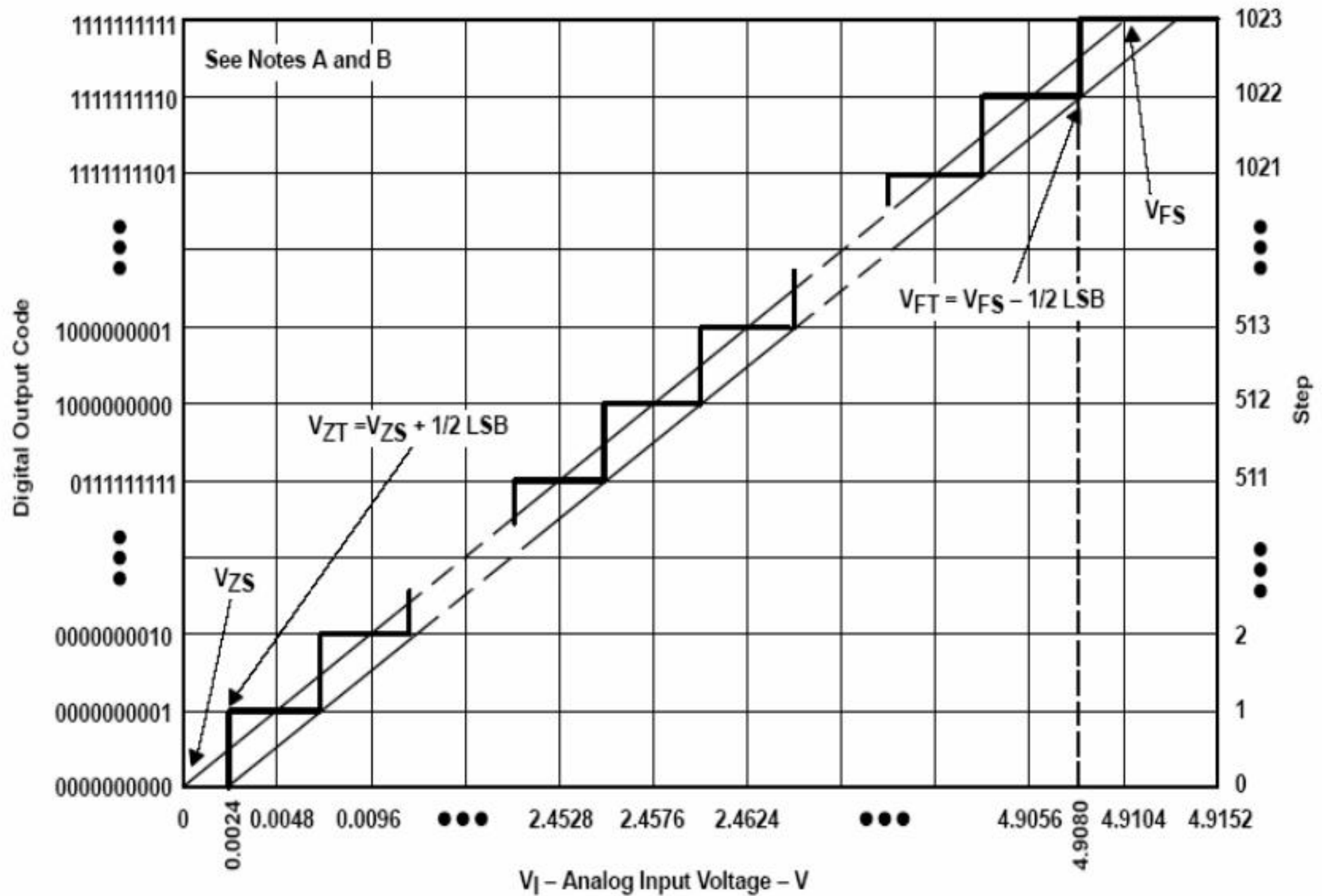
Figure 9. Timing for 10-Clock Transfer Using $\overline{CS}$

NOTE A: To minimize errors caused by noise at $\overline{CS}$, the internal circuitry waits for a setup time plus two falling edges of the internal system clock after $\overline{CS}$ before responding to control input signals. Therefore, no attempt should be made to clock in an address until the minimum $\overline{CS}$ setup time has elapsed.

图 3-4 TLC1543 的工作时序

其理想转换特性如图 3-5 所示。

APPLICATION INFORMATION



NOTES: A. This curve is based on the assumption that V_{ref+} and V_{ref-} have been adjusted so that the voltage at the transition from digital 0 to 1 (V_{ZT}) is 0.0024 V and the transition to full scale (V_{FT}) is 4.908 V. 1 LSB = 4.8 mV.
B. The full-scale value (V_{FS}) is the step whose nominal midstep value has the highest absolute value. The zero-scale value (V_{ZS}) is the step whose nominal midstep value equals zero.

图 3-5 TLC1534 理想转换特性

3.3.3.4 TLC1543 与单片机的接口

10 位分辨率的数据采集系统中用 TLC1543 实现 A/D 转换十分方便。由于单片机独特的 I/O 口操作指令和位处理功能, TLC1543 与单片机的接口非常简单。一般只需要利用 4~5 根 I/O 口线与 TLC1543 的 $\overline{CS}$ 、DATAI、CLOCK、DATDO 以及 EOC 端直接连接即可, 虽然通过 EOC 信号判断 A/D 转换是否结束, 由于 TLC1543 A/D 转换时间约为 $10\mu s$, 在一般的系统中数据采集后的处理工作常常大于 $10\mu s$, 所以 EOC 信号可以不接。TLC1543 和 AT89C2051 的接口电路如图 3-6 所示。TLC1543 的 I/O 时钟、数据输入、片选信号由 P1.5、P1.4、P1.2 提供, 转换结果由 P1.3 口串行读入。

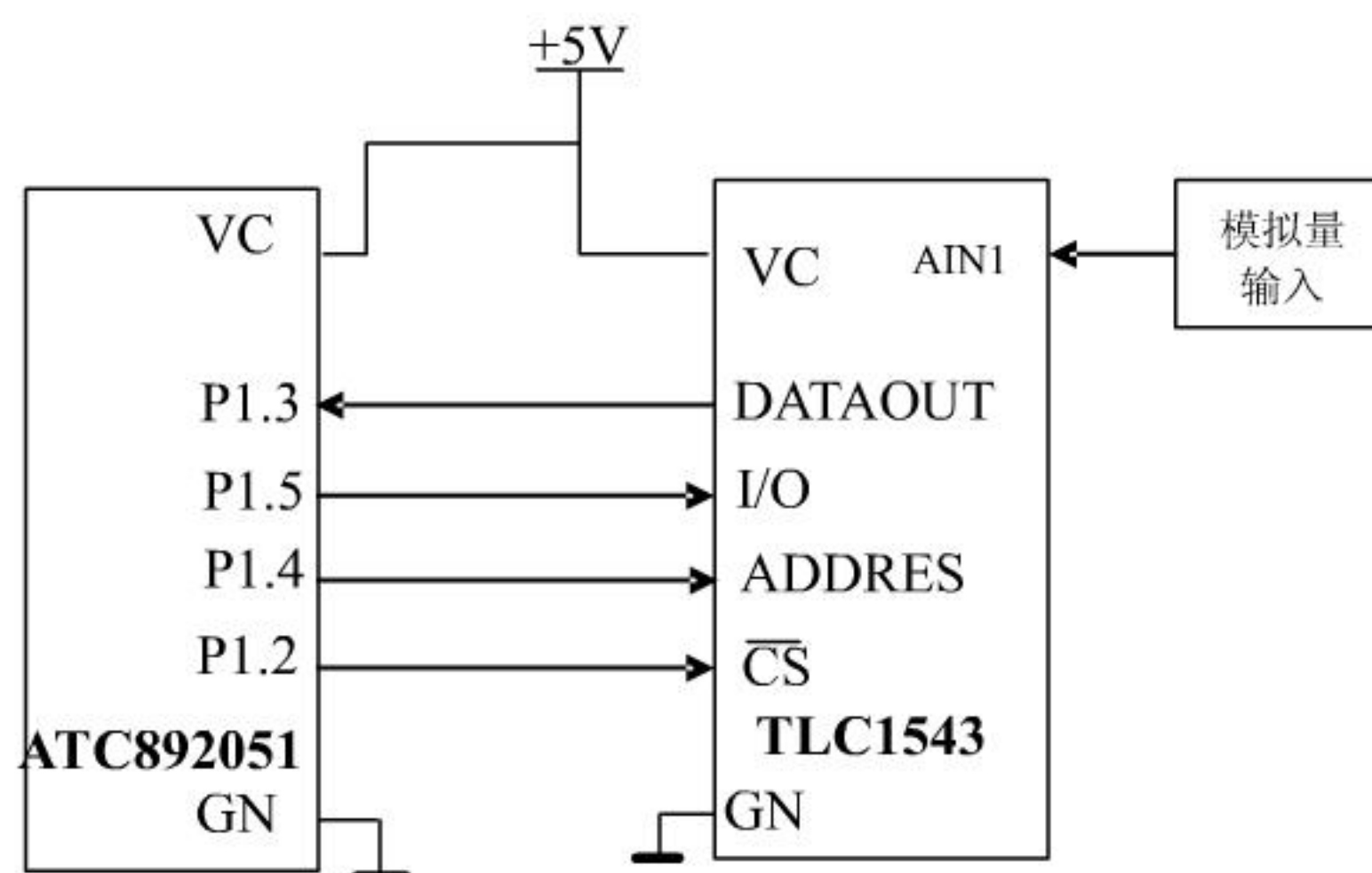


图 3-6 AT89C2051 和 TLC1543 的接口电路

3.3.4 采样模块设计

数据检测模块包括电流互感器、电压形成放大电路、整流滤波电路和信号调理电路。该控制器采用的是由简单的电流负感器采样, 经桥式整流、滤波后再经过一个简单的限幅保护电路直接将线性的电压信号送至 A/D 转换器。其限幅保护电路主要是用来保护 A/D 芯片。数据检测模块如图 3-7 所示。

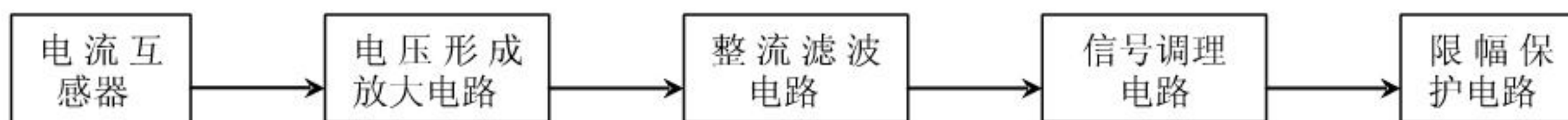


图 3-7 数据检测模块设计

3.3.5 其它电路设计

3.3.5.1 光隔电路

在单片机与继电器驱动线圈之间应增加隔离电路，如光隔等。选用 4N25 型光偶。接单片机 P1.7 口。

3.3.5.2 继电器接口电路

继电器的主要优点是用小电流的线圈控制大电流的触点。用十几毫安电流接通线圈，可使几百毫安、几个毫安的触点接通。

接口电路如图 3-8 所示。

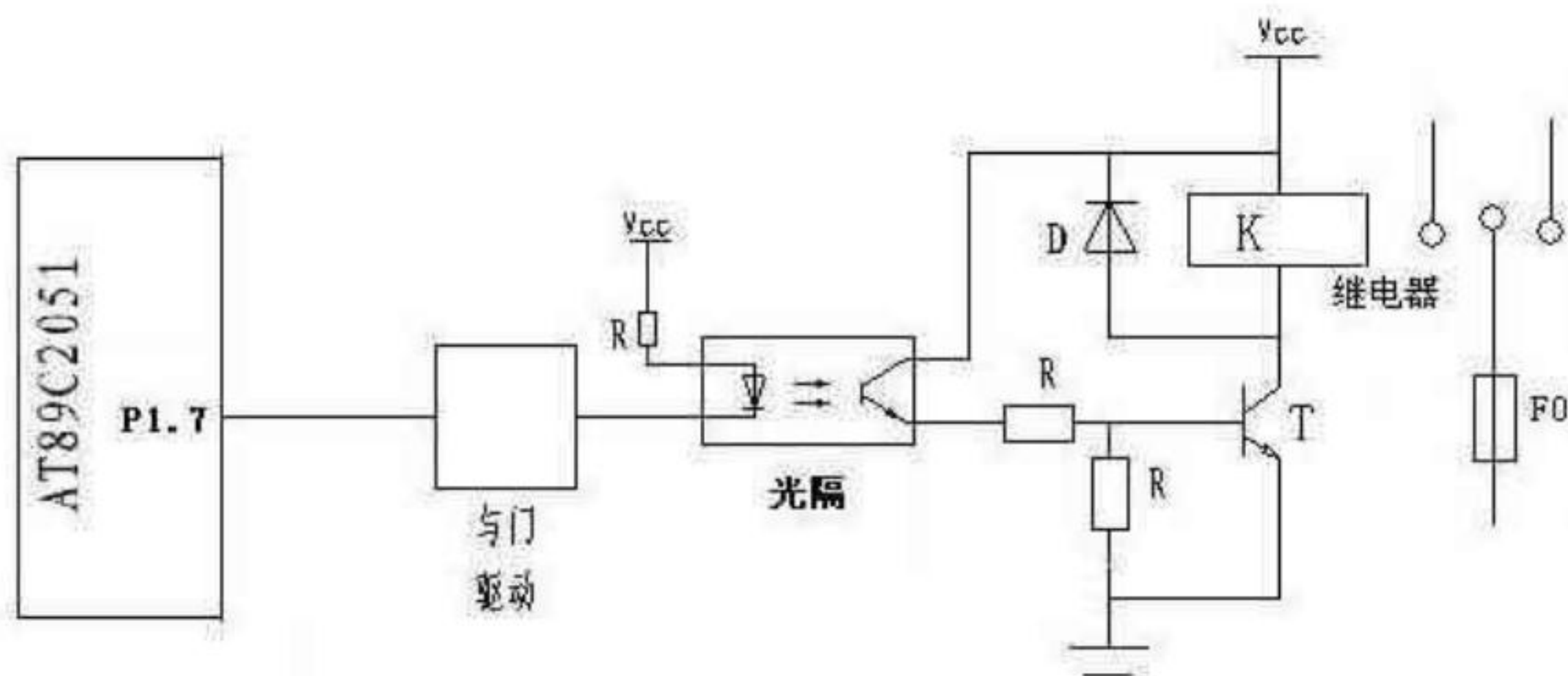


图 3-8 单片机与继电器的接口电路

注意：继电器线圈是电感性负载，所以线圈两端要并联续流二极管。

3.3.5.3 短路保护

主电路中设置了熔断器，控制器设为 10A。保护模块为断路器，起到过载、短路保护等功能。将其直接应用到控制器中，以实现宿舍电路的多重保护。

3.4 控制器 PCB 图设计

利用软件 Protel99 画 PCB 电子电路图。见附录 C

第4章 软件设计、调试及抗干扰分析

4.1 软件设计概述

当系统硬件电路设计定型后,软件设计的任务也就明确了。具体设计时最好采用模块化设计方法,根据设计要求分成相应的模块,然后分头进行,提高软件设计的工作效率。软件设计时应从以下几个方面加以考虑:

- 1) 根据软件功能要求,将系统软件分成若干个相对独立的部分,使系统软件总体结构清晰、简捷、流程合理。
- 2) 各功能程序实行模块化、子程序化。这样既便于调试、修改和扩展。
- 3) 建立正确的数学模型。它是关系到系统性能好坏的重要因素。
- 4) 为提高软件设计的总体效率,在编写应用软件之前,应绘制出程序流程图。这是程序设计的一个重要组成部分,也是决定成败的关键部分。
- 5) 要合理分配系统资源,包括 ROM、RAM、定时器/计数器、中断源等。其中最关键的是片内 RAM 分配。分配时应充分发挥其特长,做到物尽其用。当 RAM 资源规划好后,应列出一张 RAM 空间分配表,以备编程查用方便。
- 6) 注意在程序的有关位置处写上功能注释,提高程序的可读性。
- 7) 加强软件抗干扰设计,它是提高计算机应用系统可靠性的有力措施。

4.2 程序设计方法

要用计算机解决某一问题或完成一些功能,首先应按照实际问题的要求或功能,同时结合所使用计算机的特点进行分析,确定相应的算法和步骤,然后选择相应的指令,按照尽可能节省数据存放单元、缩短程序长度和运算时间三个原则编写程序,一般把这一编制程序的过程称为程序设计。

用汇编语言编写程序,一般可分为以下几个步骤:

- 1) 分析题意,确定算法:对要解决的问题进行具体的分析,找出合理的计算方法,对复杂的问题可反复研究分析并抽象出数字模型,初步确定出解题步骤。
- 2) 设计程序流程图:按已确定的算法和步骤,把这一解决问题的思想、算法和步骤具体画为程序流程图,这其中常要用到程序的一些基本结构。
- 3) 确定数据结构:合理地选择和分配内存单元以及工作寄存器。
- 4) 编写源程序:根据程序流程图,选用适当的指令和寻址方式来编制源程序。

5) 上机调试程序: 先把源程序进行汇编, 修改程序中的语法错误, 然后执行目标程序, 检查和修改设计中的错误, 对程序运行结果进行分析, 直至正确为止。在这过程中, 可逐步优化程序, 不断提高自己的设计和调试水平。

4.3 负载识别理论的发展

4.3.1 负载识别理论的作用及其主要类型

大功率电器智能识别的实现, 主要靠负载识别理论的发展。一种负载识别理论的新出现或更新都会促使智能管理系统走上一个新台阶。新理论或新算法都基于构造数学模型, 因此基础科学的发展将束缚应用科学的发展。要发展高新科技, 必须发展基础理论。目前负载识别理论以下三种:

- (1) 基于傅立叶变换的负载识别
- (2) 基于小波变换的负载识别
- (3) 基于人工神经网络的负载识别

4.3.2 基于傅立叶变换的负载识别及其在本系统中的应用

在信号分析中, 为了简化信号特征参数的提取, 经常将信号从时域表示转移到某一变换域表示, 即为信号进行线性变换。傅立叶变换是时域到频域互相转换的工具, 从物理意义上讲, 傅立叶变换的实质是把函数的波形分解成许多不同频率的正弦波的叠加和。

常见的学生公寓的用电负载类型为阻性负载(如白炽灯、电炉、热得快)、感性负载(如日光灯)、容性负载(如风扇)、非线性负载(如计算机)以及它们混合时的负载。一般情况下, 电压波形为正弦波, 即标准电压; 当一些学生公寓由于年代久远, 供电设备和线路老化, 线路容量不足时, 会使电压波形产生畸变, 成为非正弦波, 称为削顶电压。相同负载在标准电压和削顶电压下, 它们的波形是不同的。要实现以上两种电压波形下负载类型的识别, 以限制大功率性负载, 同时允许使用计算机负载, 就要对电器负载的类型进行识别。

在实际负载识别过程中, 由于学生公寓电器类型多样, 直接对各种电器负载的功率进行识别比较困难, 因此可转而识别计算机的功率, 再从总功率中减去计算机的功率, 所得到的功率就是包含日光灯、白炽灯及违章电器等负载的功率。因为日光灯、白炽灯等的功率较小, 因此可设定一阈值, 当非计算机的功率大于这一阈值时, 即认为是违章用电。

4.3.3 基于小波变换的负载识别及其在本系统中的应用

小波分析是一种信号的时间—尺度(时间—频率)分析方法, 它具有多分辨率分析的特点。原则上讲, 传统上使用傅立叶分析的地方, 都可以用小波分析取代。小波分析优于傅里叶变换的地方是, 它在时域和频域同时具有良好的局部化性质。

直接从原始样本来进行分类是无目的的，这是由于对分类来说，重要的不是一个模式的完整描述，而是导致区别不同类别模式的那些“选择性”信息的提取，也就是说，特征提取的主要目的就是尽可能集中表现显著类别差异的模式信息。另一个目的则是尽可能缩小数据集，以提高识别率，减少计算量。

目前，小波分析已经广泛地应用于信号处理、地震勘探、语音识别与合成、机械故障诊断与监控等科技领域。由负载的电压和电流信号的波形可推测，电压和电流信号是由几个不同频率的正弦信号所组成，因此可以采样小波变换来进行负载识别。

利用小波变换方法可识别用电负载中是否含有计算机负载、是单纯的计算机负载还是计算机与其它电器的混合负载，还可以更进一步的识别出计算机的台数。此种方法的不足之处是不能区分是正弦电压还是非正弦电压，需要通过观察电压和电流的波形得到电压的工作状态。

4.4 本系统采用的负载识别算法

采用傅立叶算法或小波分析算法时，计算量庞大，对处理器要求较高，在误差要求不是很高的场合可以采用差值比较的算法。

通过试验比较，一般小功率的电器，其使用时电流波形比较接近正弦函数，但像计算机一类的功率稍大的电器，其电流波形变化比较显著，两者的波形比较如图 4-1 和图 4-2 所示：

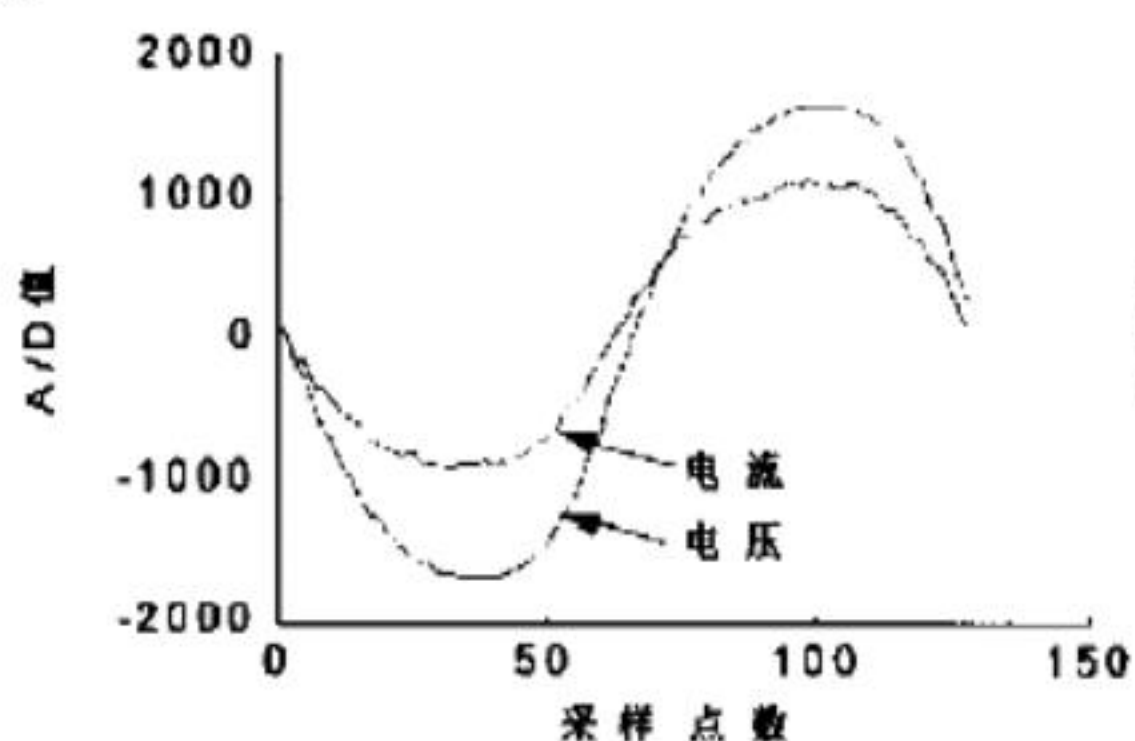


图 4-1 白炽灯的波形

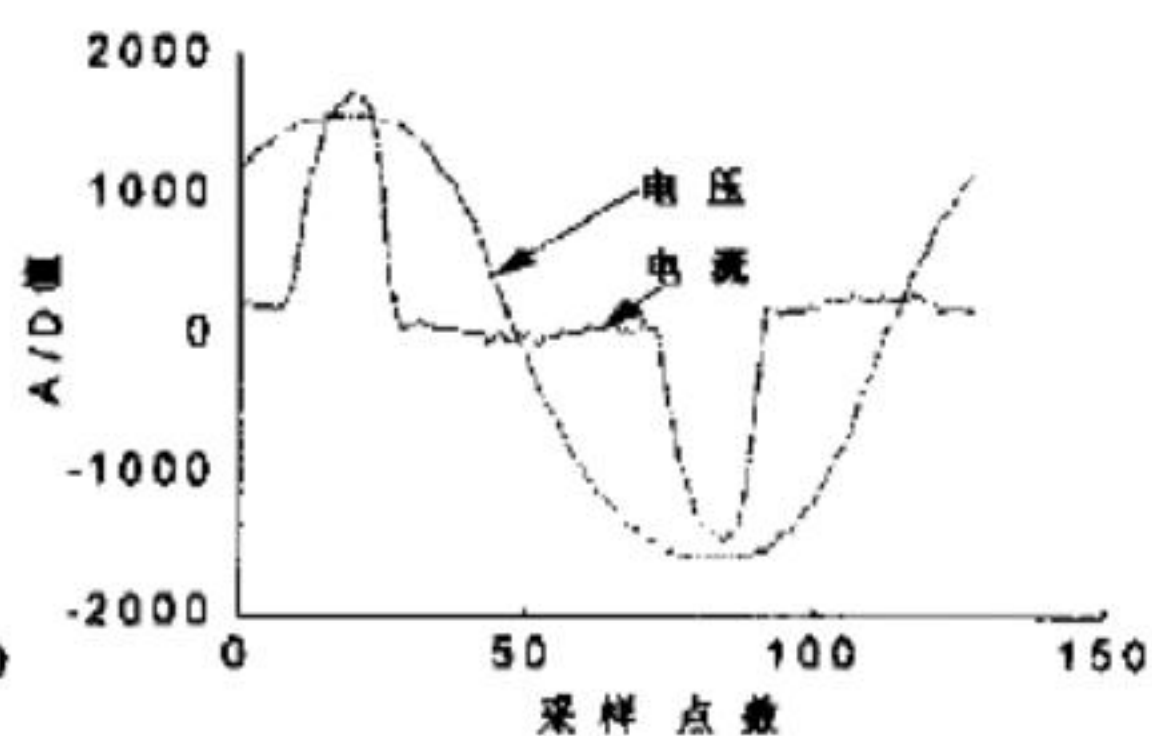


图 4-2 计算机的波形

我们可以检测其在出现突变值后的电流变化，如出现一个极大值之后的一段时间连续出现大值则可判断其为大功率负载，而普通电器的电流变化较平缓，在一些情况下虽然也可能出现电流的跳变，但干扰过去以后，其电流的变化又恢复平缓变化。

4.4.1 算法实现

在本系统中，我们通过电流互感器和其外围电路的参数选择，使 A/D 转换的结果与负载功率的大小形成每增加一个二进制单位对应负载增加 1W 的对应。根据采用的 A/D 芯片的转换特性，查芯片资料得到模拟的电压量与二进制数字量的对比值。假设单个电器的最大限定值为 300W（可调），算法步骤如下：

系统软件每 50mS（可自定义）对负载电流进行 1 次采样，连续采样 18 次（可自定义）为一次计数周期。每次 A/D 转换后的值都与一个采样计数周期的第一次（18 次中的第一次）的采样值求差，如果差值达到 300W 对应的二进制值则将大值标志位记 1，重新计数采样次数，并将上次 A/D 转换的电流数据做为暂时设定值，即为一个计数周期内的第一次采样值，继续采样电流的数据。接下来的 A/D 转换值如果大于暂时设定值，则将计数器加 1，若在一个计数周期内计数器的值达到上限设定值（本系统设为 10 次），则判断目标房间使用了大功率电器，计数器和大值标志位清零，驱动继电器断电；若在一个计数周期内计数器的值没有达到上限设定值，则其电流变化可能为干扰信号，计数器和大值标志位清零，重新检测电流的变化。

4.4.2 本系统负载识别算法的另一种思路

由上述的差值比较法可知，大功率负载的电流变化幅值变化较剧烈，则其功率的平均变化率将大于一般功率的负载，我们选取功率波形的斜率进行分析，且理想化其功率变化与时间成正比关系（因为在采样时我们选取波形的两点的幅值大小进行比较，可近似看成以穿过这两点的直线的斜率作为其变化趋势的度量）。

假设目标房间的单个电器的上限使用功率为 300W（可调），房间里不同时间内使用了两个功率和为 500W 的电器（其中一个电器功率可为零，即使用了 500W 的大功率电器），我们理想化目标房间的功率变化曲线如图 4-3 所示。

取一个采样计数周期 T （如计数 18 次，每隔 50ms 采样一次，即 $T = 18 \times 50\text{ms} = 900\text{ms}$ ，可自定义），对 P_1 曲线进行 n 次采样，则两条曲线的斜率为：

$$K_0 = \frac{300}{T},$$

$$\overline{K_1 + K_2 + \dots + K_n} = \frac{\sum_{i=1}^n \frac{P_{i+1} - P_1}{T/n}}{T}。$$

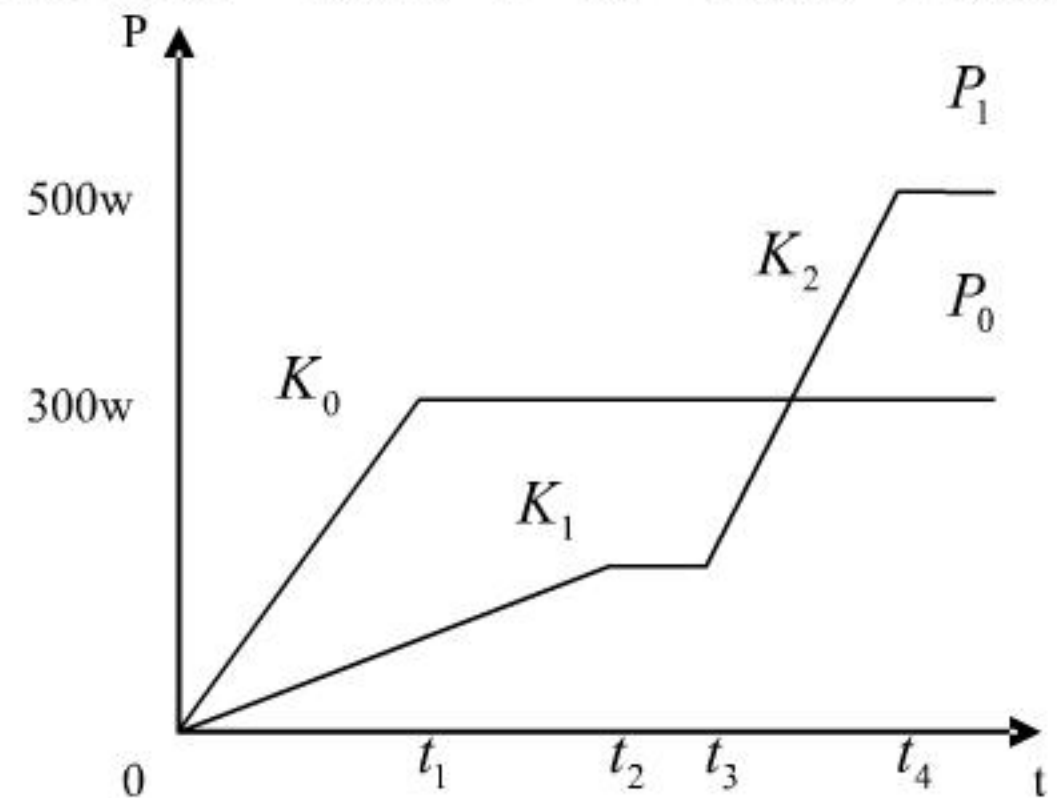


图 4-3 功率曲线

当一个采样计数周期 T 内 $\overline{K_1 + K_2 + \dots + K_n} \geq K_0$ 时，则判定目标房间使用了大功率电器。

算法中采样的计数周期 T 不能设置得太小，其取值过小将出现小功率电器功率变化过快引起系统误判，同时也不能太大，根据实际调试来设定。

4.5 软件编程

4.5.1 程序算法分析及软件设计方案确定

4.5.1.1 程序算法分析

从 TLC1543 进行 A/D 转换的结果经 I/O 口传入单片机内部存储单元中，采样次数随存储单元大小而定（控制器设置为 18 次），检测功率突变次数（控制器设置为 10 次）达到设定值，单片机即刻驱动继电器动作。调用延时 15 秒灯闪子程序后，进行重合闸次数（控制器设为 3 次）判断是否需要手动复位重合闸。

算法中的关键点：

- ①主程序开始延时 1 秒，仅可能保障单片机正常稳定工作。
- ②堆栈指针 SP 赋初值，防止程序 PC 值冲突导致错乱。
- ③设置标志位，是程序的关键，能有效地引导程序的正常运行。
- ④对采样数据进行减法运算，从而计算出功率突变值。
- ⑤通过数据差值与功率下限设定值的比较来判断是否为大功率电器。

在整个程序中，通过设置标志位，是编程变得简单了：

- i. 定时到标志位 20H.0，定时信号到才进行数据采集及其处理；
- ii. 设置标志位，给二字节减法程序和比较程序带来了方便，如标志位 20H.1、20H.2 及 F0。

4.5.1.2 系统软件设计方案确定

根据设计要求，首先要确定软件设计方案，即确定该软件应该完成哪些功能；其次是规划这些功能需要分成多少个功能模块，以及每一个程序模块的具体任务是什么。模块的划分有很大的灵活性，但也不能随意划分。划分模块时应遵循下述原则：

- 1) 每个模块应具有独立的功能，能产生一个明确的结果。
- 2) 模块之间的控制参数应尽量简单，数据参数应尽量少。控制参数是指模块进入和退出的条件及方式，数据参数是指模块间的信息交换（传递）方式、交换量的多少及交换的频繁程度。
- 3) 模块长度适中。模块语句的长度通常在 20 条~100 条的范围较合适。模块太长时，分析和调试比较困难，失去了模块化程序结构的优越性；模块太短则信息交换太频繁，也不合适。

根据模块的划分原则，我们将该本系统的程序划分成七个模块，如图 4-4 所示。

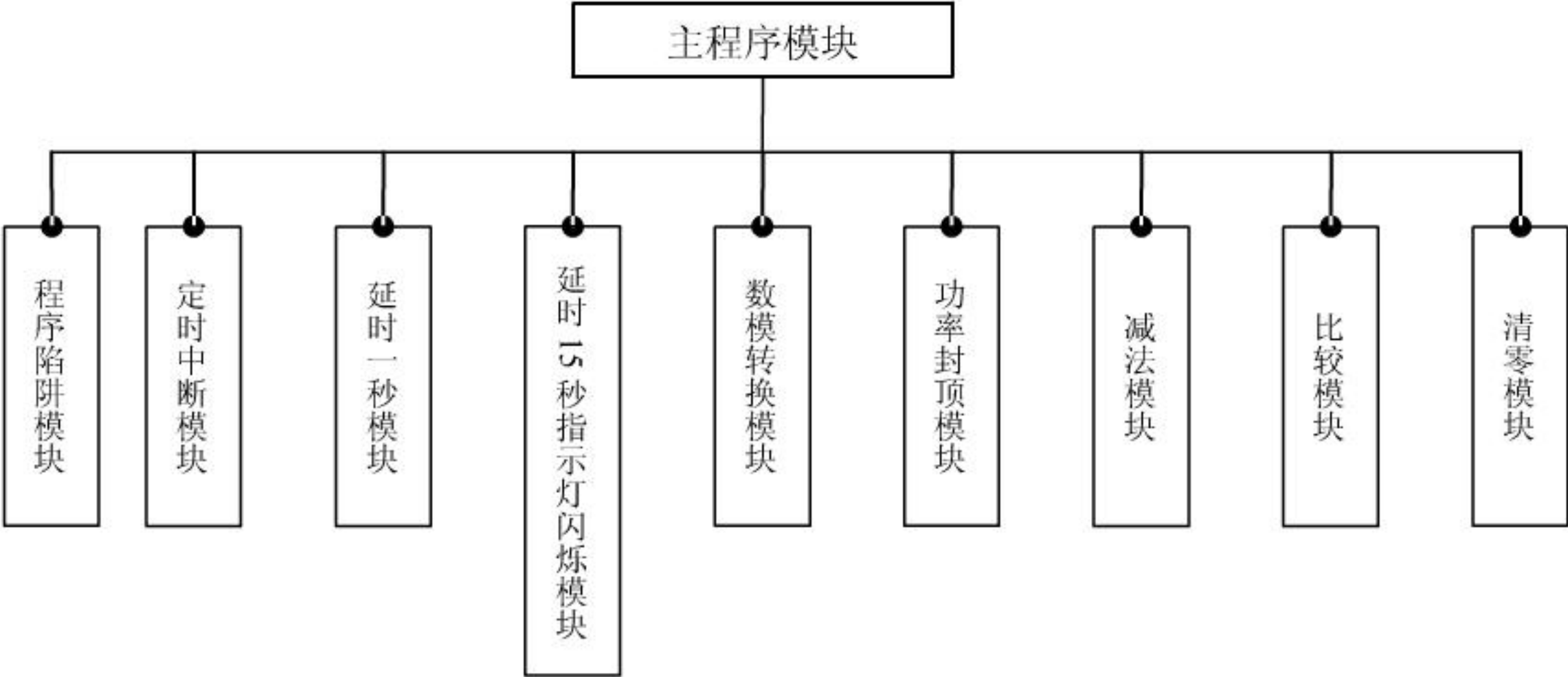


图 4-4 系统软件设计的模块

4.5.2 内存分配表

一般在程序设计前应对内存进行合理的分配，这样既可使得编程时心中有数，又可充分利用单片机内部 RAM 有限的存储空间。本课题的内存分配：数据存储器地址空间分配见表 4-1 所示。

表 4-1 片内部分数据存储区

7FH		F0	出现大值标志位
6AH	堆栈指针 SP 初值		
69H	暂存 A/D 数据区		
68H			
67H	暂存差值		
66H		20H.2	高 2 位减法出现负值标志位
65H	存设定值	20H.1	低 8 位减法出现负值标志位
64H		20H.0	定时到标志位
63H	存储 A/D 数据单元		位寻址区
62H			
			第三工作寄存器区
			第二工作寄存器区
			第一工作寄存器区
		R7	none
		R6	A/D 转换子程序中作计数用
41H		R5	none
40H		R4	重合闸次数
33H		R3	大于设定值次数计数
32H		R2	取 A/D 转换数据次数计数
31H		R1	none
30H		R0	数据单元地址指针

4.5.3 主程序设计及流程图

主程序主要完成智能识别大功率电器及驱动继电器动作的功能。其中智能识别功能的实现是通过对电流定时进行采样,从 TLC1543 进行 A/D 转换的结果经 I/O 口输入单片机内部存储单元中,检测功率突变达到设定值次数和维持时间,根据判断控制继电器动作。延时 15 秒灯闪子程序后,进行重合闸。并判断是否需要手动复位重合闸。主程序流程图如图 4-5 所示。

程序开始前,先字符赋值、位定义,然后写上中断入口地址及 MAIN 汇编地址等。然后调用延时 1 秒子程序(有足够的时间躲过断电后重新冲来电时可能存在的尖峰电流,及保证了单片机的稳定工作)。定时器 T0 初始化及开中断允许,工作寄存器置初值(包括

A/D 通道号、计数用寄存器置处置#00H，把设定值写进 65H、66H 单元)。

调用 A/D 转换子程序：第一次调用读出无效数据，因为第一次读出的数据为随机的通道号转换而来的，即读出的数据为随机数，无效。

启动定时，SETB TR0，等待中断。

判断定时到标志位来对 A/D 数据根据预定的算法进行处理。

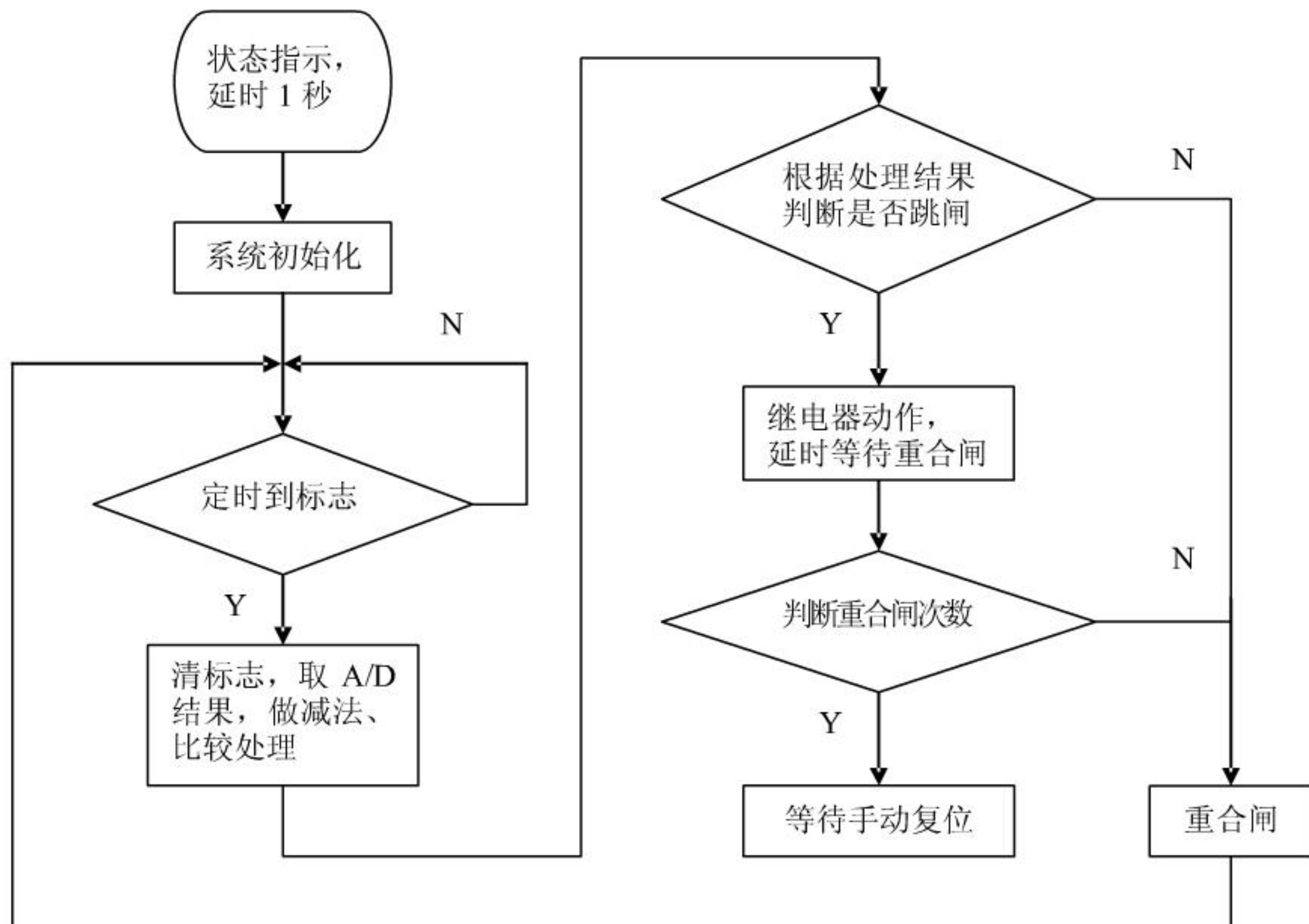


图 4-5 主程序流程图

4.5.4 主要子程序及其调用设计

在实际的程序设计中，经常会出现某段程序或某种结构的程序多次出现，如果编程中每遇到这样的操作都编写一段该程序，会使编程十分烦琐，可读性也不强。常常把这些基本操作功能编制成独立的程序段，并尽量使其标准化，需要时通过指令进行调用。这样的程序段称为子程序。调用子程序的程序称为主程序或调用程序。使用子程序的过程称为子程序调用。

在主程序中需要执行该程序段时，可用一条子程序调用指令（如 LCALL），当程序执行到该条指令，就转到子程序中完成整个子程序的操作，然后由子程序的最后一条返回指令 RET，返回到原来的程序继续执行下去。

子程序的使用需要注意以下几点：

① 子程序的第一条指令的地址称子程序的首地址或称为入口地址，必须用（符号）标号标明，可用英文的缩写或拼音的缩写，以便调用指令正确、方便地调用。

② 现场保护：一般在子程序的第一部分把主程序（或调用程序）的关键空间、中间结果保护起来，常用压栈保护，以免子程序执行过程中改变了这些关键值，但要记住，在子程序返回前要将它们恢复，同时注意出栈恢复的顺序。

③ 子程序的末尾用 RET 返回指令结束，以便正确返回调用程序继续执行下去。

④ 参数传递：在调用子程序前，主程序应先把有关参数（即入口参数）放到某些约定的位置，子程序在运行时可以从约定的位置/单元得到有关的参数。同样，子程序在运行结束返回前，也应该把运算结果（出口参数）送到约定的位置/单元。在返回主程序后，主程序可以很方便从这些地方得到需要的结果。这就是参数传递。

实际编程时，可采用多种参数传递的方法，例如累加器或寄存器、堆栈等进行参数的传递。

本课题中的子程序是负载识别算法中的主要环节，首先是 A/D 采样子程序，采样数据的精确度影响着整个系统的准确性；其次是 SUB 减法子程序，是对采样数据进行求差；由 COMP 比较子程序来对差值与设定的下限值进行比较，并设置了标志位。

4.5.4.1 A/D 采样子程序设计和流程图

在 A/D 转换程序中，我们将 1543 设置为 10 位数据，MSB（高位）先导输出，采用 0 通道。编程时注意到第一次读出的数据为随机数，为无效数据应在主程序中先被读出。在定时器服务程序中调用 A/D 采样子程序。数据的传送采用位传送指令，编程简单。A/D 采样程序流程图如图 4-6 所示。部分 A/D 采样子程序如下：

```
AD10:      CLR      AD_CS
MSB:        MOV      R6,#04H      ; 采样数据前 4 位
MS1:        MOV      C,AD_DOUT
            RLC      A
            MOV      AD_DIN,C
            SETB     AD_CLK
            NOP
            NOP
            CLR      AD_CLK
```

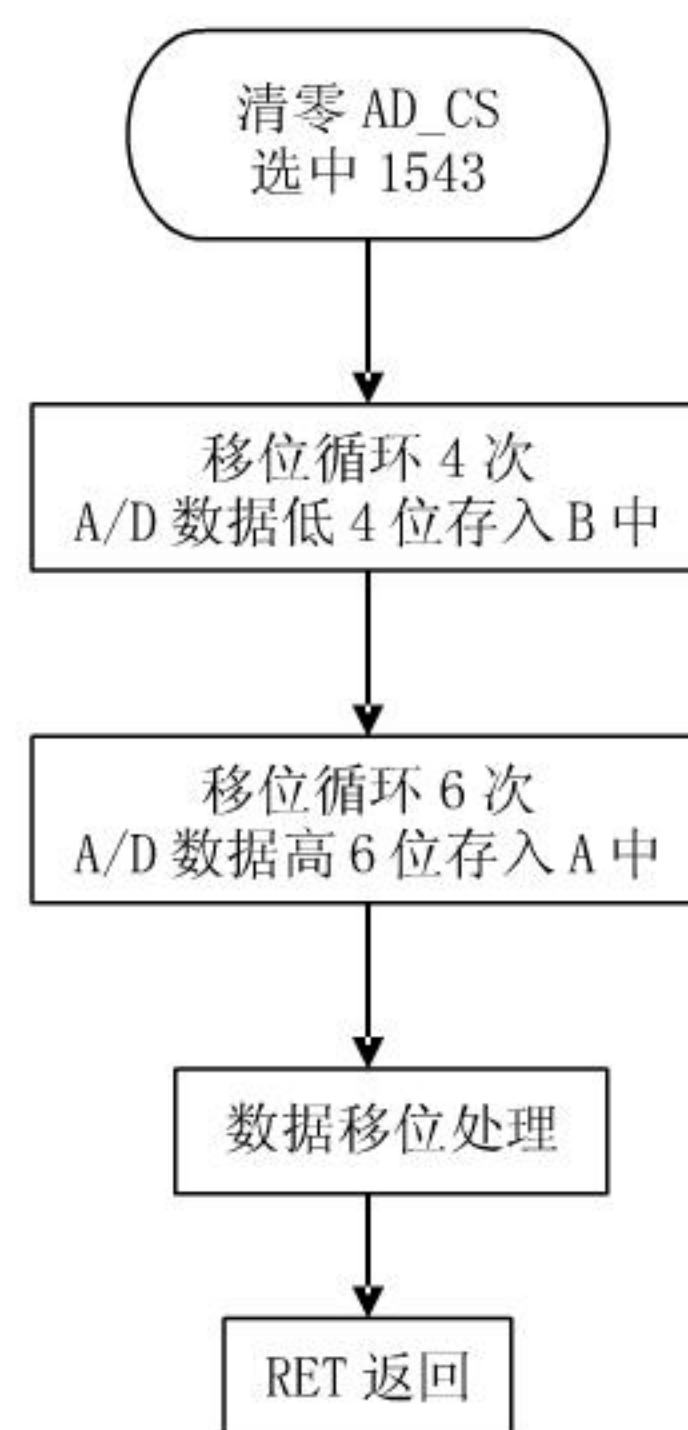


图 4-6 A/D 采样程序流程

```

        DJNZ R6,MS1
        MOV B,A
        MOV A,#00H
LSB:    MOV R6,#06H          ; 采样数据前 6 位
LS1:    MOV C,AD_DOUT
        RLC A
        SETB AD_CLK
        NOP
        NOP
        CLR AD_CLK
        DJNZ R6,LS1
        SETB AD_CS
        MOV A.6,C
        MOV C,B.1
        MOV A.7,C
        XCH A,B
        RR A
        RR A
        ANL A,#03H
        XCH A,B
        RET ;
    
```

4.5.4.2 SUB 减法子程序设计和流程图

总体思路是对采样数据与前一次采样周期的最后一次采样做差运算。A/D 数据为两字节，因此减法为两字节减法运算，先做低 8 位减法，再做高 2 位减法。

注意：①考虑到减法出现负值的情况，故设置标志位（这里为 20H.0）来区分差值的正负性，以便作不同处理。

②低字节做减法有借位时，也特别设置了标志位（这里为 20H.1）来进行标志。

SUB 减法子程序流程图如图 4-7 所示。

4.5.4.3 COMP 比较子程序设计和流程图

先判断减法子程序中是否出现负值的情况，若有负值出现，程序直接返回；若为正值，就把差值与功率下限的设定值（这里设为#68H）进行比较，出现大值，就给计数器 R3 加 1，

并设置标志位 F0，否则程序返回。

注意：①设置出现大值标志位 F0，这样才能在主程序中正确判断是否出现了大值。

②当第一次出现大值时，上次 A/D 采样数据应及时存入减法程序中的减数单元（这里定义为 3EH、3FH）。

COMP 比较子程序流程图如图 4-8 所示。

减法子程序清单如下：

```

SUB:      SETB      20H.1
          SETB      20H.2
          MOV       A,@R0
          CLR       C
          SUBB      A,3FH
          JNC       MARK1
          CLR       20H.1
MARK1:    MOV       67H,A
          DEC       R0
          MOV       A,@R0
          JB        20H.1
          SETB      C
MARK2:    SUBB      A,3EH
          JNC       MARK3
          CLR       20H.2
MARK3:    MOV       66H,A
          INC       R0
          RET
  
```

比较子程序清单如下：

```

COMP:     JNB       20H.2,RETN3
          MOV       A,64H
          CJNE      A,66H,NEXT
          MOV       A,65H
          CJNE      A,67H,NEXT
NEXT:     JC        WORK
          CLR       F0
          SJMP      RETN3
WORK:     CJNE      R3,#00H
          DEC       R0
          DEC       R0
          MOV       69H,@R0
          DEC       R0
          MOV       68H,@R0
          MOV       R0,#3EH
          MOV       @R0,68H
          INC       R0
          MOV       @R0,69H
          MOV       R2,#01H
          INC       R0
          INC       R0
WORK1:    SETB      F0
          INC       R3
RETN3:    RET
  
```

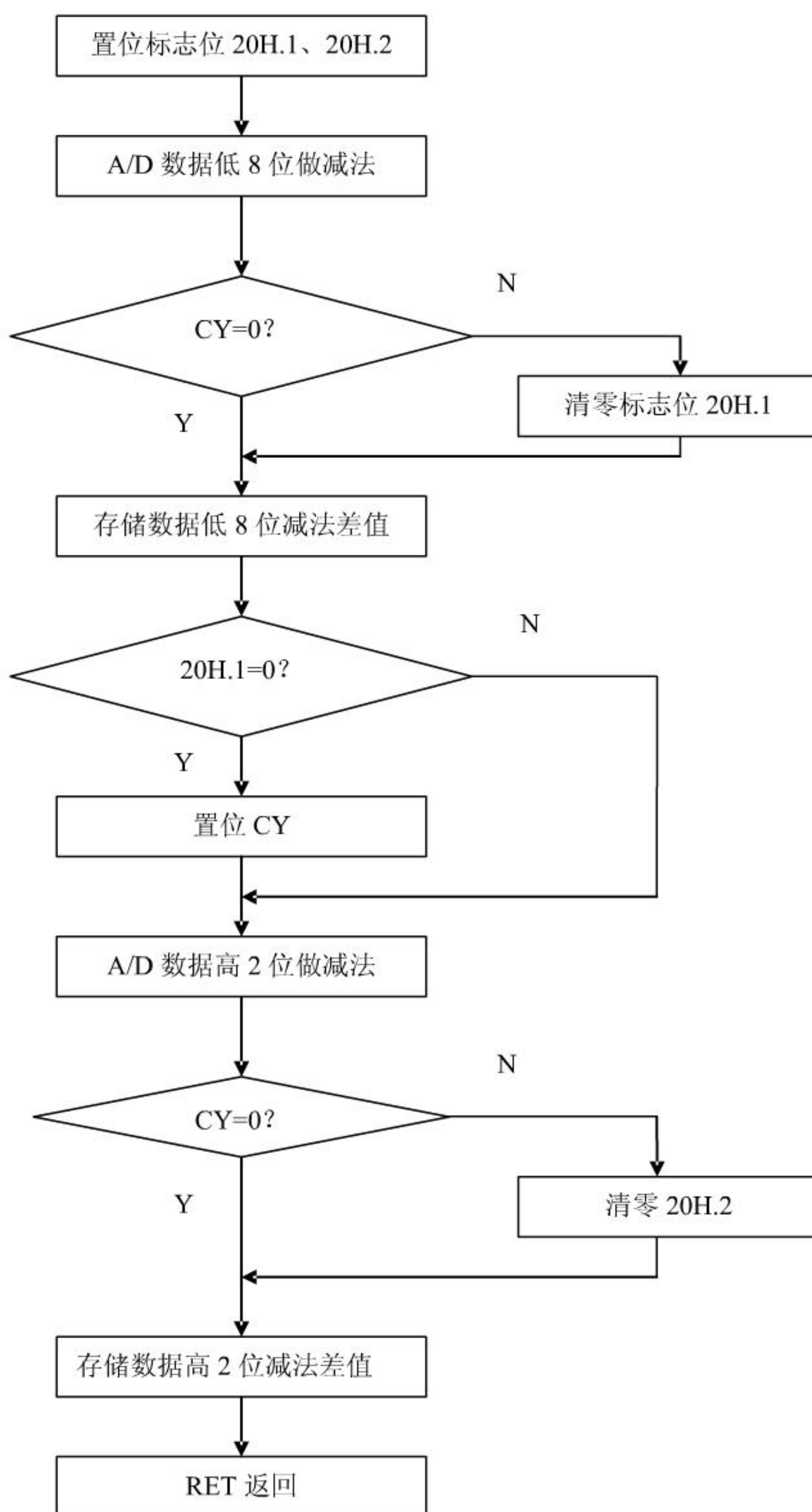


图 4-7 SUB 减法子程序流程图

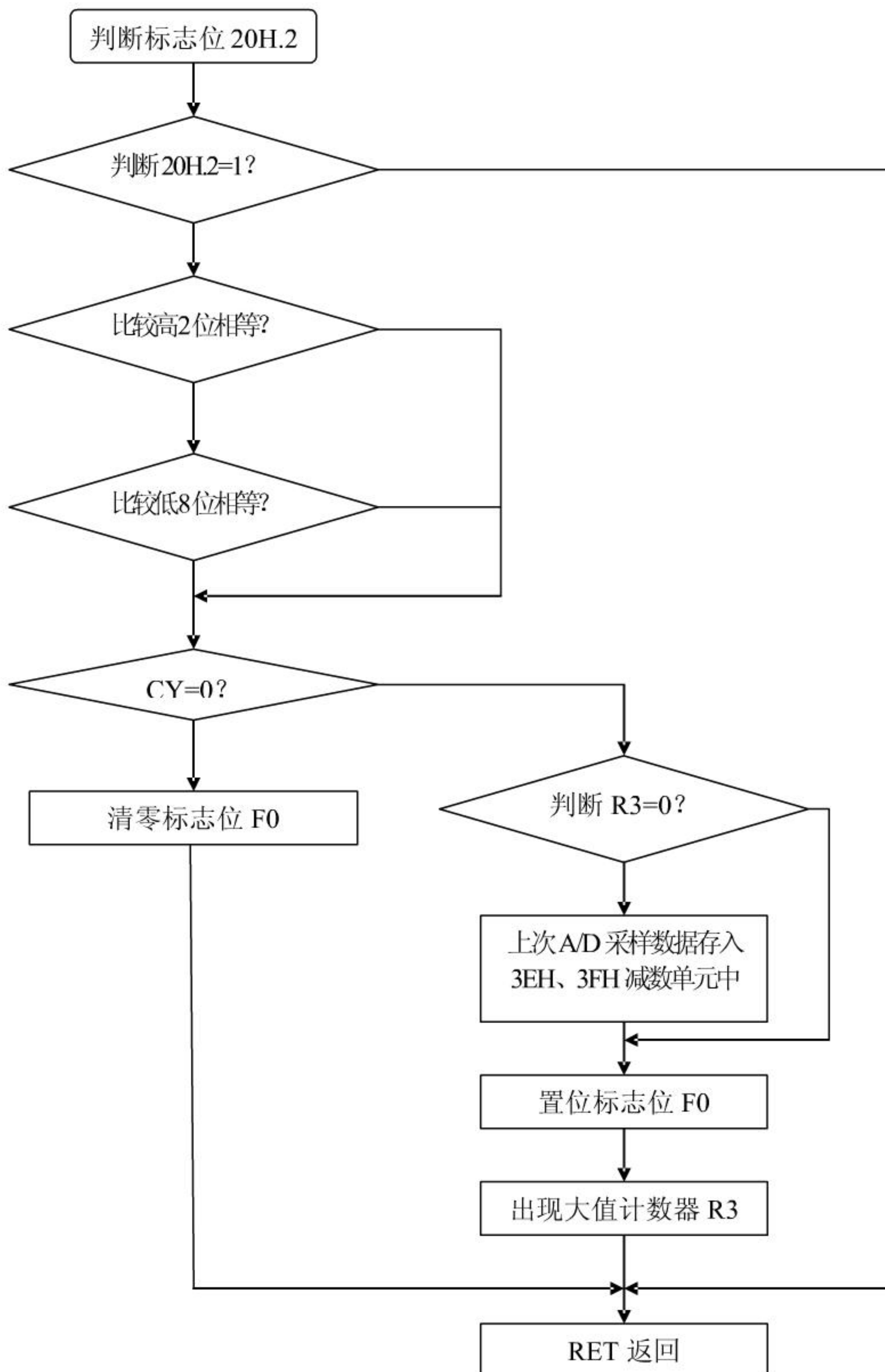


图 4-8 COMP 比较子程序流程

4.6 系统的调试

4.6.1 应用系统的调试与运行.

在系统硬件设计完成后,需要制作电路板(PCB板)和进行元器件的采购。PCB板做好后,进行系统的组装,即元器件的焊接,然后要进行系统调试。

单片机应用系统的调试包括硬件调试和软件调试。由于硬件和软件是紧密交织在一起的,硬件调试和软件调试并不能完全分开,所以要将两者结合起来进行调试和调整。许多硬件错误是在软件调试过程中被发现的。一般方法是先排除明显的硬件故障,再进行软、硬件综合调试。本节主要介绍如何利用仿真器进行调试、利用编程器进行程序固化和应用程序的运行。

(一) 应用系统的调试方法

当设计者完成了设计与组装工作后,在进行联机仿真调试之前,先做好应用系统调试前的准备工作:

1) 首先用眼睛或用万用表直接检查线路板各处是否有明显的断路、短路的地方,尤其是要注意电源是否短路。元器件的安装是否正确。

2) 完成上述检查后,先空载上电(未插芯片),检查线路板各管脚及插件上的电位是否正常,特别是单片机管脚上的各点电位(若有高压,联机调试时会通过仿真线进入仿真系统,损坏有关器件)。若一切正常,将芯片插入各管座,再通电检查各点电压是否达到要求,逻辑电平是否符合电路或器件的逻辑关系。若有问题,断电后再认真检查故障原因。排除明显的硬件故障后,就可以进行联机仿真调试了。

1. 用单片机开发系统调试电路功能

用户的源程序经过汇编后,生成的目标文件必须经过仿真调试,才能固化到应用系统的程序存储器 ROM 中。

单片机仿真器是帮助设计者进行应用系统调试的专用工具,并通过串口与计算机相连接,构成单片机开发系统。仿真器的软件具有自己的行编辑功能,用来编辑用户的汇编语言源程序,同时它具有编译功能,以及单步、跟踪、设置断点和连续运行程序的调试能力。从实质上讲,应用系统的调试是对软硬件的综合调试,一方面要排除软件错误,同时进一步解决硬件部分的遗留故障。调试时可以将软件模块一个一个分开进行,再将它们连在一起统调。

仿真调试时,需将仿真器的仿真插头插入到应用系统空出的 CPU 插座中,将开发系统主机和仿真器相连;通电启动后,用户便可以在监控状态下,通过主机键盘输入各种操作、控制、运行命令,对应用系统进行在线仿真调试。仿真器提供的调试命令有很多,如要详

细了解请查阅有关仿真器用户手册。

2. 各功能模块的调试

在调试的过程中，应把整个应用系统按其功能分成若干模块，如系统显示模块、键盘输入模块、A/D 模块等。针对不同的功能模块，编写一小段测试程序，并借助于万用表、示波器、逻辑笔等仪器来检查硬件电路的正确性。

利用“测试程序”对系统的运行调试很有帮助。特别是对经验较少的开发者，尤其重要。

如果简单的测试程序通过，说明硬件没有什么大问题；如果简单程序模块不能运行，则说明硬件有问题。这时就应该首先将你的硬件化简成最小系统，排除硬件故障后再运行“测试程序”。并逐步增加软件模块和硬件模块，反复实验。

（二）应用系统程序的固化

用户目标程序经过仿真调试通过后，就可以进行固化了。固化的方法有两种：一是用仿真器配备的固化功能，二是用专门的编程器进行。用专门的编程器进行程序固化操作简单，使用方便。

（三）用户程序的运行

经过仿真调试的用户目标程序固化到 EPROM 后，将其插入到样机的器件插座中，接通电源，就可运行样机。

在一切正常后，样机即研制完成。一般情况下，目标样机都能正常工作。若脱机工作不正常，则应考虑样机上的复位电路是否有故障，系统晶振是否起振等。

应用系统的调试、开发是一件复杂而细致的工作，只能给出大致步骤，经验还需在实践中不断积累。

4.6.2 控制器的应用

控制器对电流互感器的线性度要求较高，要求互感器的工作点必须取在线性区域而且互感器在额定电流范围内不会饱和，否则系统将可能误动作或不动作。

电压形成电阻的阻值与分压电路中电位器大小的配合也非常重要。电压形成电阻如取得太小，采集到的信号电压将较小，这样分压电位器必然调到较大的值从而使信号输入内阻较大，影响采样的准确度。相反分压电阻取的太大有可能导致负感器饱和，同样会影响采样的线形度。此外在低负荷时（小于 200W）整流桥的管压降也将对采样的线形度产生一定影响。但只要电压形成电阻与分压电位器的大小配合得当就能将误差控制在规定范围内。

根据控制器的设计要求，其采样滤波电容不能取得太大，否则将不能灵敏的反映采样信号的跳变。经反复测试取 2UF 为最佳。

控制器使用方法简单，安装方便无须另设线路。设有两个指示灯，黄灯亮为系统电源

指示，红灯亮为跳闸指示。自动重合闸时红灯闪烁，跳闸次数超过设定次数时将不再自动重合闸，红灯长亮等待手动复位合闸。

该控制器已应用于某校的部分学生宿舍，通过实际应用该控制器能够长期稳定工作，并且性能稳定、动作可靠。常规电器都能在总功率设定范围内正常使用。当有超过允许使用单个电器的上限值的大功率电器使用或负载总功率超过设定值时，电路将立即跳闸，延时后自动重合闸，并且只有在主电路负载重新回到正常范围内后自动重合闸才能成功，否则在自动重合闸失败超过设定次数就只能通过手动复位合闸。

4.7 系统的抗干扰分析及采取的措施

4.7.1 应用系统的抗干扰分析

单片机应用系统由于体积小，抗干扰能力强，多用于生产现场。因此，容易受到现场各种信号的干扰，直接影响到系统的可靠性。因此，应用系统的抗干扰设计已经成为设计人员关注的重要课题。

对单片机应用系统造成的干扰有电网干扰、传输线干扰、空间电磁波干扰和机内干扰。由于干扰源不同，产生的影响也不同。单片机应用系统大多采用交流供电，来自电网的干扰主要是浪涌电压和电磁干扰。因此，当电网电压长时间欠压、过电压和电网频率不稳定时，会影响单片机应用系统的性能指标。传输线干扰是在输入、输出口连接线上形成的干扰，亦称为通道干扰。通道干扰又分为前向通道和后向通道干扰，前向通道的干扰会使输入的模拟信号失真、数据采集误差加大、数字信号出错。后向通道受到干扰后，干扰信号会通过输出通道反串控制系统，使系统发生“死锁”，或控制误差加大及控制失常。电磁干扰的能量是以电磁波的形式在空中传播的，电磁波不仅会严重干扰设备的正常工作，还会造成程序失控、控制失灵，以及直接危及人体健康。系统内部干扰信号包括电磁继电器产生的火花放电、自激振荡、噪声电压等，单片机系统内部受到干扰后，会使总线上的数字信号发生错乱，从而引发一系列无法预料的后果，并会传递下去，形成一系列错误。导致程序失控、死循环。解决系统的抗干扰问题一般有两个途径，一是硬件抗干扰技术，二是软件抗干扰技术。在许多情况下，应用系统的抗干扰不可能完全依靠硬件来解决。采用软件抗干扰设计，往往成本低，见效快，起到事半功倍的效果。

4.7.2 抗干扰采取的措施

应用系统的干扰设计可以从硬件和软件两方面来考虑：

设计硬件电路时，可以从电路设计、器件选择、电路板元器件布置与地线设计等方面注意抗干扰问题。首先，电路设计时要注意电平匹配。如 TTL“1”电平是 2.4~5 伏，“0”电平是 0~0.4 伏，而 CMOS“1”电平是 4.99~5 伏，“0”电平是 0~0.01 伏。因此，当 CMOS

器件接受 TTL 输出时，其输入端就要加电平转换器或上拉电阻，否则，CMOS 器件就会处于不确定状态；其次是模拟地和数字地要分开；强弱电可通过光电耦合器进行隔离；另外 CMOS 电路不使用的输入端不允许浮空，否则会引起逻辑电平不正常，易接受外界干扰产生误动作。在设计时可根据实际情况，将多余的输入端与正电源或地相连接。单片机应用系统的硬件抗干扰设计是整个抗干扰设计的主要部分，也是抗干扰设计的基础，它必须为软件抗干扰设计提供良好的条件。用硬件来解决这些干扰的方法也很多，如屏蔽、隔离、滤波、接地、保护电路等方法。

在印刷电路板设计中，各元器件的摆放位置要合适，布局设计要将强、弱电路严格分开，尽量不要把它们设计在一块印刷电路板上；电源线的走向应尽量与数据传递方向一致；接地线应尽量加粗，在印刷电路板的各个关键部位应配置去耦电容；

地线设计是一个不容忽视的问题。在微机应用系统中，地线结构大致有系统地、机壳地（屏蔽地）、数字地、模拟地等。在设计时，数字地和模拟地要分开，分别与电源端地线相连；当系统工作频率小于 1MHZ 时，屏蔽线应采用单点接地；当系统工作频率在 1M~10MHZ 时，屏蔽线应采用多点接地。

软件的抗干扰设计，是应用系统抗干扰设计的一个重要组成部分。在许多情况下，应用系统的抗干扰不可能完全依靠硬件来解决。采用软件抗干扰设计，往往成本低，见效快，起到事半功倍的效果。单片机在正常运行中，难免会遇到外界干扰使 CPU 发生纷乱引起“死机”。程序“跑飞”后往往将一些操作数当作指令码来执行，从而引起整个程序的纷乱。消除程序“跑飞”的方法之一，是采用“指令冗余”使“跑飞”的程序恢复到正常状态。另一种方法就是采用所谓的“软件陷阱”。“软件陷阱”是一条指令，强行将捕捉的程序引向一个指定的地址，在那里有一段专门处理错误的程序。这段程序的入口地址为 ERROR，下面五条指令即组成一个“软件陷阱”。

NOP

NOP

NOP

NOP

LJMP ERROR ；转到预先设计的入口执行

“软件陷阱”一般安排在未使用的中断向量区，未使用的大片 EPROM 空间，表格和程序区。

第 5 章 联网管理软件及操作界面的设计

5.1 概述

要实现网络管理需要设计上位机管理软件。

由下位机收集的数据经总线送到上位机,再进行各种电量计算,上位机由计算机构成,而计算机的串口采用 RS232 接口标准,需转换才能与 RS485 总线相连,具体做法是先利用 MAX202E 将 RS232 标准转换成 TTL 电平,再利用 MAX3089 将 TTL 电平转换为 RS485 标准,这样每级的数据汇总后经过转换器可以顺利的集中统计到计算机中。

计算机与各收集点的单片机构成主从式多机通讯系统,主机收集数据时按顺序广播特定从机的地址信息,每个从机都能收到这个地址信息,然后与自己的地址进行比较,相同就发送数据,不同则继续等待,这样每一时刻主机只与一台从机通讯,实行轮回的数据监控。

各数据汇总到计算机后,由管理软件计算各种电量数据,其主要功能有:

- (1)查询功能 可进行房间电量、欠费告警、超限用电、欠费关断等各种查询。
- (2)登录功能 实现交费登录、入住登录等。
- (3)系统参数设置 设置电费单价、设置(修改)房间基础用电量、设置瞬时用电量最高限值、设置月累计用电量最大限值、设置电费交费告警限值等。
- (4)通讯功能 主要实现对各房间巡检、电源通断、各种数据(基础电量、购买电量等)传送。

5.2 管理系统

管理系统主要由用户输入/输出部分、数据采集器、通讯模块、单元控制器、主控计算机组成。

用户输入部分主要包括有各种计费仪表,其中有水表、电度表、热能表等,还有火灾报警器等。

数据采集器对各个用户的信号进行采集、转换存储和向主机传输,包括有微处理器、数模转换、检测电路、供电电路等。

单元控制器是数据采集器与控制计算机相互连接的桥梁,根据控制计算机的指令来读取各个数据采集器的数据,通过串行总线 RS485 与数据采集器通讯,通过 RS232 与控制计算机通讯。

控制计算机选用性能稳定的兼容或品牌机即可。

各系统多采用 RS485 总线是因为 RS485 收发器采用平衡发送和接收,具有抑制共模干扰的能力,其具有很高的灵敏度,能检测低至 200mV 的电压,故传输的信号能在千米以外得到恢复。RS485 的接受器的输入负载小,可以允许一个传送器驱动多个接收器;而且处于禁止状态的输出的漏电流小,可以允许多个传送器的输出并联。它的这种通讯距离远、抗干扰能力可靠性高和成本低的特点,使得它在分散目标监控系统和集抄系统等场合获得了广泛的应用。

电平的转换有多种方式,可以采用标准的 RS232C 电平转换器 MC1488 和 MC1489 来完成,也可以采用三极管等分立件组成电路进行电平转换,还可以利用集成的 RS232 接口芯片进行电平转换,如 MAXIM 公司的 MAX232 等。综合各种方式,利用分立件组成电路,故障点较多,因此,利用集成的接口芯片是最好的选择。

计算机操作系统选用 WINXP 较稳定,管理软件设计方面,由于 VISUAL BASIC 在通讯方面的优越性,可以成为系统的首选开发工具。ACCESS 数据库用户界面友好,使用方便,具有级联删除修改功能,比较适合系统要求。

5.3 VB 程序设计基础

Visual Basic 是美国微软公司1991年推出的,它提供了开发Microsoft Windows 应用程序的最迅速、最简捷的方法。它不但是专业人员得心应手的开发工具,而且易于被非专业人员掌握使用。

5.3.1 Visual Basic 概述

Visual Basic 是以结构化 BASIC 语言为基础,以事件驱动作为运行机制的新一代可视化程序设计语言。其中,Visual 指的是开发图形用户界面(GUI)的方法,它不需要编写大量代码去描述界面元素的外观和位置,而只要把预先建立的对象添加到屏幕上即可;Basic 指的是 BASIC (Beginners All-Purpose Symbolic Instruction Code)语言,它是计算机技术发展史上应用最广泛的语言之一。Visual Basic 在原有 BASIC 语言的基础上进一步发展,综合运用了 BASIC 语言和新的可视化设计工具,既具有 Windows 所特有的优良性能和图形工作环境,又具有编程的简易性。

5.3.2 界面设计

在 Windows 环境下运行的应用程序,虽然不是所有的用户界面看上去或操作起来都一样,但是这些界面由许多的特点。他们都包括一些基本要素,如菜单、工具栏、状态栏,而在应用程序的操作过程中又往往要打开一些对话框,如保存文件对话框、设置字体对话框、设置颜色对话框等等。使用 Visual Basic 开发 Windows 环境下的应用程序,同样离不

开这些常用的界面要素的设计。

(1) 菜单设计

Windows 环境下几乎所有的应用软件都通过菜单实现各种操作。使用 Visual Basic 设计应用程序时，当操作比较简单，且操作命令较少时，可以直接通过控件来执行；而当要完成比较复杂且命令选项比较多的操作时，使用菜单操作会更为方便。

菜单的基本作用有两个：一是提供人机对话的界面，将应用程序的各种操作分组显示在界面上，由用户方便地进行选择；二是管理应用程序，控制各种功能模块的执行。

菜单可以分成两种基本类型：下拉式菜单和弹出式菜单。例如，启动 Visual Basic 后，单击“文件”菜单所显示的就是下拉式菜单，而在窗体上单击鼠标右键打开的菜单为弹出式菜单。下拉式菜单通常通过单击菜单标题打开。弹出式菜单通常在某一区域通过单击鼠标右键打开。

(2) 工具栏设计

(3) 状态栏设计

通常，Windows 应用程序都具有状态栏，状态栏一般显示在窗口的底部，用于显示应用程序当前的运行状态、系统状态等，并提供一些操作提示。使用 Visual Basic 提供的状态栏控件 StatusBar 可以很容易地设计出具有 Windows 风格的状态栏。

(4) 多文档界面设计

(5) 对话框设计

在 Windows 应用程序中，对话框是用户和应用程序交互的主要途径，它常用来向用户提供输入数据或进行选择的界面，或者向用户显示一些提示信息。例如 Microsoft Word 中的“页面设置”对话框、“查找和替换”对话框等。

结束语

当前，家电数字化、楼宇配电及电量抄表自动化正在普及，学生公寓用电安全管理智能化却出现了空档。而随着生活质量、电器科技水平的不断提高，家庭常用电器越来越多，用电量越来越大，用电安全问题的解决就更加迫切。

大功率电器智能识别与用电安全控制系统，填补了这一空档。该系统可实现学生公寓各房间非法用电自动识别、短路、过流保护、查询用电情况和欠费自动提示等功能，采用本系统可使公寓用电管理自动化、智能化，提高公寓管理水平，实现安全、节能、增效。大功率电器智能识别与用电安全控制器能自动检测线路负荷，能对控制线路中安全量、最大量和负荷极限量三种情况做出反应和提示，并在多次警示无效的情况下自动切断总电源，避免了频繁更换线路保险器件的麻烦，更从根本上杜绝了由线路超载而引发的火灾。

大功率电器智能识别与用电安全控制器成本低，易于形成产品，组成网络系统实现对学生公寓用电的智能控制。能广泛应用于集体生活区，特别是广大中小学、大中专高等院校的学生宿舍。目前已在部分高校使用，性能可靠、稳定。

参考文献

- [1] 王迎旭. 单片机原理与应用. 北京: 机械工业出版社. 2004 年
- [2] 刘国荣. 单片微型计算机技术. 北京: 机械工业出版社, 1996 年
- [3] 吴金戌,沈庆阳,郭庭吉. 8051 单片机实践与应用. 清华大学出版社. 2001 年 8 月
- [4] 杨振江,杜铁军,李群. 流行单片机实用子程序及应用实例. 西安电子科技大学出版. 2002 年 4 月
- [5] 高峰. 单片微机应用系统设计及实用技术. 电子工业出版社. 2004 年 4 月第 1 版
- [6] 房小翠. 单片微型计算机与机电接口技术. 国防工业出版社. 2001 年 11 月 30 日
- [7] 苏凯, 刘庆国,陈国平. MCS-51 系列单片机系统原理与设计. 冶金工业出版社.2003 年
- [8] 何立民. 单片机应用系统设计. 北京航空航天大学出版社. 1990 年
- [9] 孙琳. 公寓用电智能管理控制系统. 辽宁师专学报. 2004 年三月第 6 卷第一期. 文章编号:1008- 5688 (2004) 01-0087-03
- [10]刘宁宁. 集群式用电智能管理系统. 石家庄铁道学院. 学位论文 2004.2.1
- [11]吕进华,傅玉朋,李明浩. 基于 RS—485 的公寓用电集中管理. 大连民族学院学报. 2004 年 9 月第 6 卷第 5 期
- [12]张重,张道信,姜宝林. 学生宿舍总线式限电计量计算机管理系统. 吉林建筑工程学院学报. 2001 年 6 月第 2 期. 文章编号:100920185 (2001) 0220037203
- [13]滕世进. 智能配电管理系统. 电子世界. 2005 年第 5 期 P36
- [14]农建波,韦江维,班世炳. 一种新型的用电管理系统. 电子技术. 2000 年第 8 期
- [15]刘宁宁,亢海伟,高蒙. 小波分析在负载识别中的应用. 河北省科学院学报. 2004 年第 21 卷

致 谢

本毕业设计的全部工作是在王迎旭教授的悉心指导下完成的，论文的字里行间凝聚着老师的汗水和心血。老师渊博的学识、严谨的治学态度使作者受益匪浅，将激励着本人在今后的工作中不断进取，永不怠慢。

值此论文完成之际，特向王老师表示衷心的感谢！

刘荔鑫

2005 年 6 月

附录 A 部分程序

；程序清单

```

AD_MAX      EQU      64H          ； 定义设定值
AD_DATA     EQU      40H          ； 定义 A/D 数据存储单元
AD_CS       BIT      P1.2         ； 2051&1543 的位定义
AD_CLK      BIT      P1.5
AD_DOUT     BIT      P1.3
AD_DIN      BIT      P1.4

ORG         0000H                ； 中断程序陷阱
SJMP        MAIN

ORG         0003H
LJMP        ERROR

NOP
NOP

LJMP        ERROR

ORG         000BH
LJMP        TODO                 ； T0 中断

NOP
NOP

LJMP        ERROR

ORG         0013H
LJMP        ERROR

NOP
NOP

LJMP        ERROR

ORG         001BH
LJMP        ERROR

NOP
NOP
    
```



```

        LJMP      ERROR
        ORG       0023H
        LJMP      ERROR
        NOP
        NOP
        LJMP      ERROR

MAIN:    CLR      P3.7          ; 电源指示引脚
        CLR      P3.7
        LCALL    ONE          ; 调用延时 1 秒子程序
        MOV      TMOD,#01H    ; 定时器初始化
        MOV      TH0,#03CH    ;#03CH    ;50ms ; #0F8H    ;2ms ;
                                #0B1H    ;20ms
        MOV      TL0,#0BAH    ;#0BAH    ;#3AH    ; #0EAH
        SETB     ET0          ; 开中断允许
        SETB     EA
        MOV      SP,#6AH      ; 堆栈指针赋初值
        MOV      SP,#6AH      ; 指令冗余，防止程序跑飞
        MOV      20H,#0FFH    ; 位地址所在单元赋值
        SETB     P1.6          ; 继电器动作端口、动作指示
        SETB     P1.7
        SETB     P1.1
        SETB     P1.6          ; 指令冗余
        SETB     P1.7
        SETB     P1.1
        CLR      20H.0        ; 清零定时到标志
        CLR      F0           ; 清零出现大值标志
        CLR      PSW.4        ; 定义工作寄存器工作组别
        CLR      PSW.3
        MOV      R0,#AD_DATA  ; A/D 存储单元地址
        MOV      R1,#00H      ;none
        MOV      R7,#00H      ;none

```

	MOV	R2,#00H	; 采样计数器
	MOV	R3,#00H	; 超出设定值计数器
	MOV	R4,#03H	; 重合闸次数
	MOV	R6,#00H	
	MOV	3EH,#00H	; 减法子程序中的减数
	MOV	3FH,#00H	
	MOV	64H,#00H	; 设定值赋初值
	MOV	65H,#068H	
	MOV	A,#00H	; 取通道号
	LCALL	AD10	
	MOV	R0,#AD_DATA	; 第一次调用无效 A/D 数据
	SETB	TR0	; 启动定时
	NOP		
	NOP		
IMP:	JNB	20H.0,RET2	; 判断定时到没?
	JNB	20H.0,RET2	
	CLR	20H.0	; 定时到清标志位
	CLR	20H.0	
	MOV	A,#00H	; 取 A/D 通道号
	LCALL	AD10	; 调用 A/D 转换子程序
	MOV	@R0,B	; 存储 A/D 转换结果
	INC	R0	
	MOV	@R0,A	
	INC	R2	; A/D 采样次数加 1
	NOP		
	NOP		
	LCALL	TOP	; 调用封顶子程序
	NOP		
	NOP		
	LCALL	SUB	; 调用减法子程序
	NOP		
	NOP		

	LCALL	COMP	; 调用比较子程序
	NOP		
	NOP		
	JNB	F0,CONTINU	; 判断出现大值标志位
	JNB	F0,CONTINU	
	CLR	F0	; 出现大值, 标志位清零
	CLR	F0	
	CJNE	R3,#0AH,CONTINU	; 判断出现大值次数是否达到 10 次
	CJNE	R3,#0AH,CONTINU	
	CLR	P1.7	; 达到 10 次, 继电器动作, 指示灯 闪烁
	CLR	P1.6	
	CLR	P1.1	
	DJNZ	R4,GO	; 判断重合闸次数
	CLR	P1.7	; 继电器动作, 等待重合闸
	CLR	P1.6	
	CLR	P1.1	
	SJMP	\$	
	NOP		
	NOP		
	NOP		
	NOP		
	LJMP	ERROR	
GO:	LCALL	DELAY	; 调用延时 15 秒子程序
	SETB	P1.7	; 继电器重合闸
	SETB	P1.6	
	SETB	P1.1	
	MOV	@R0,#00H	; 清零减法子程序中的减数单元数据
	DEC	R0	
	MOV	@R0,#00H	
	INC	R0	
	SJMP	RETN1	

```

NOP
NOP
NOP
NOP
LJMP    ERROR
CONTINU: CJNE    R2,#12H,RET1      ; 判断 A/D 转换次数是否达 18 次
          CJNE    R2,#12H,RET1
RET1:    LCALL    CLEAR           ; 调用清零子程序
          SJMP    RET2
          NOP
          NOP
          NOP
          NOP
          LJMP    ERROR
RET1:    INC      R0              ; A/D 结果存储单元地址加 1, 准
                                   备下一次采样
RET2:    SJMP    IMP             ; 跳转 IMP 重新检查定时到标志
          SJMP    IMP
          NOP
          NOP
          NOP
          NOP
          LJMP    ERROR

ERROR:   LJMP    MAIN           ; ERROR 子程序, 跳转 MAIN 主
                                   程序
          RET                  ; S=4
          NOP
          NOP
          NOP
          NOP

```



```

                                LJMP      ERROR

TODO:    MOV      TH0,#03CH      ;#03CH ;50ms ; #0F8H ;2ms  ; #0B1H  ;20ms
          MOV      TL0,#0BAH      ;#0BAH  ; #3AH  ;  #0EAH
          SETB     20H.0          ;定时到标志位置 1
          SETB     20H.0
          RETI                      ; S=7
          NOP
          NOP
          NOP
          NOP
          LJMP     ERROR

ONE:      MOV      30H,#50          ; 延时一秒子程序
LOP:      MOV      31H,#100
LOP1:     MOV      32H,#98
          DJNZ     32H,$
          DJNZ     31H,LOP1
          DJNZ     30H,LOP
          RET                      ; T=1S
          NOP
          NOP
          NOP
          NOP
          LJMP     ERROR

DELAY:    CLR      EA
          CPL      P1.6            ; 延时 15 秒指示灯闪烁子程序
          MOV      33H,#07H
LOOP:     CPL      P1.6
          LCALL    ONE
          SETB     P1.6

```

```

        LCALL    ONE
        DJNZ     33H,LOOP
        SETB     EA
        RET                               ; T=17S
        NOP
        NOP
        NOP
        NOP
        LJMP     ERROR

AD10:   CLR      AD_CS                    ; A/D 转换子程序
MSB:    MOV      R6,#04H
MS1:    MOV      C,AD_DOUT
        RLC      A
        MOV      AD_DIN,C
        SETB     AD_CLK
        NOP
        NOP
        CLR      AD_CLK
        DJNZ     R6,MS1
        MOV      B,A
        MOV      A,#00H
LSB:    MOV      R6,#06H
LS1:    MOV      C,AD_DOUT
        RLC      A
        SETB     AD_CLK
        NOP
        NOP
        CLR      AD_CLK
        DJNZ     R6,LS1
        SETB     AD_CS
        MOV      C,0F0H
    
```



```

MOV    0E6H,C
MOV    C,0F1H
MOV    0E7H,C
XCH    A,B
RR      A
RR      A
ANL     A,#03H
XCH    A,B
NOP
NOP
RET      ;a/d=B,A ;st=103us
NOP
NOP
NOP
NOP
NOP
LJMP    ERROR

TOP:    DEC    R0                ; 功率封顶子程序
MOV     A,@R0
CJNE    A,#02H,TOP1    ;>=#00FF turn  ;850W
CJNE    A,#02H,TOP1
TOP1:   JC     TOP2
JC      TOP2
SETB    F0
INC     R3
TOP2:   INC    R0
RET      ; S=15
NOP
NOP
NOP
NOP
LJMP    ERROR

```

```

SUB:      SETB      20H.1          ; 减法子程序
          SETB      20H.2
          MOV       A,@R0
          CLR       C
          SUBB      A,3FH
          JNC       MARK1
          JNC       MARK1
          CLR       20H.1
          CLR       20H.1
MARK1:    MOV       67H,A
          DEC       R0
          MOV       A,@R0
          JB        20H.1,MARK2
          JB        20H.1,MARK2
          SETB      C
MARK2:    SUBB      A,3EH
          JNC       MARK3
          JNC       MARK3
          CLR       20H.2    ;MARK-
          CLR       20H.2
MARK3:    MOV       66H,A
          INC       R0
          RET              ; Smax=30
          NOP
          NOP
          NOP
          NOP
          LJMP      ERROR

COMP:     JNB       20H.2,RETN3    ; 比较子程序
          JNB       20H.2,RETN3
    
```


	MOV	A,64H
	CJNE	A,66H,NEXT
	CJNE	A,66H,NEXT
	MOV	A,65H
	CJNE	A,67H,NEXT
	CJNE	A,67H,NEXT
NEXT:	JC	WORK
	JC	WORK
	CLR	F0
	CLR	F0
	SJMP	RETN3
	NOP	
	NOP	
	NOP	
	NOP	
	LJMP	ERROR
WORK:	CJNE	R3,#00H,WORK1
	CJNE	R3,#00H,WORK1
	DEC	R0
	DEC	R0
	MOV	69H,@R0
	DEC	R0
	MOV	68H,@R0
	MOV	R0,#3EH
	MOV	@R0,68H
	INC	R0
	MOV	@R0,69H
	MOV	R2,#01H
	INC	R0
	INC	R0
WORK1:	SETB	F0
	INC	R3

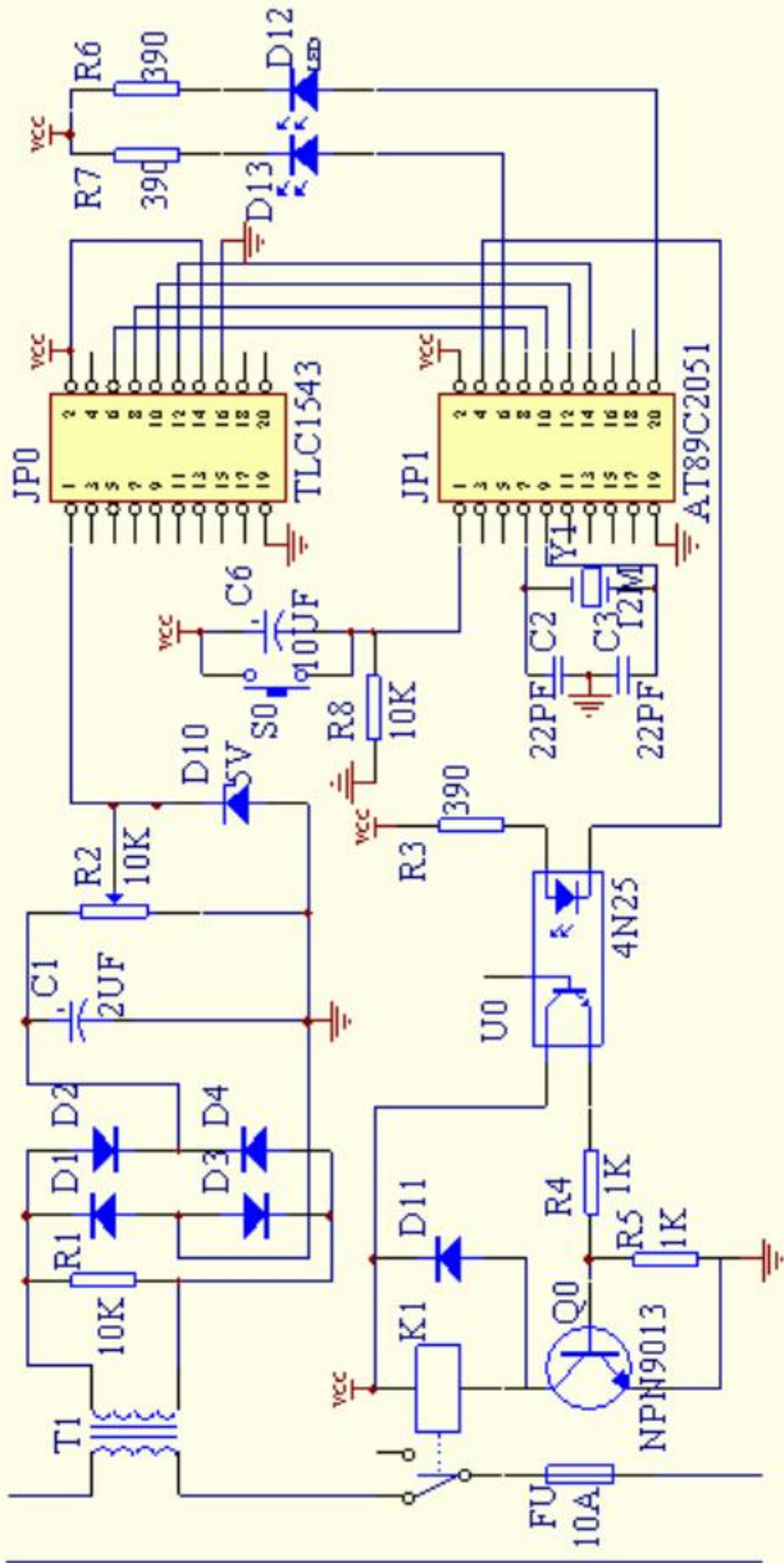
```

RETN3:    RET                                ; Smax=41
          NOP
          NOP
          NOP
          NOP
          LJMP    ERROR
    
```

```

CLEAR:    MOV     R2,#00H                    ; 清零子程序
          MOV     R3,#00H
          MOV     3FH,@R0
          DEC     R0
          MOV     3EH,@R0
          MOV     R0,#AD_DATA
          MOV     R2,#00H
          MOV     R3,#00H
          MOV     R0,#AD_DATA
          RET                                ; S=13
          NOP
          NOP
          NOP
          NOP
          LJMP    ERROR
          END
    
```


附录 B 主电路原理图



附录 C 主电路 PCB 图

规格:100×90

