

陶瓷学院科技艺术学院

本科生毕业设计（论文）

中文题目： 辊道窑温度智能控制系统

英文题目： INTELLIGENT CONTROL SYSTEM OF ROLLER
KILN TEMPERATURE

院 系： 科技艺术学院

专 业： 10 自动化

姓 名： 徐长文

学 号： 201030455115

指导教师： 朱 永 红

完成时间： 2014 年 5 月 23 日

摘 要

陶瓷产品的烧成主要通过陶瓷窑炉完成，评价陶瓷窑炉的先进性、优良性和经济性主要有烧成质量、单位能耗、自动化水平、使用寿命、环保水平等方面的指标。用单片机控制的陶瓷辊道窑能很好达到以上几个技术指标的要求。本设计加热元件为天然气阀门，均匀地布置于窑体，形成一个恒温带，由单片机进行温度控制及故障检测，可以有效地提高产品质量和生产效率。本文所设计的利用单片机实现辊道窑温度控制系统，具有人工输入温度设定、调节参数设定、加热循环周期参数设定等功能。

关键词： 辊道窑 温度控制 PID 整定 AT89C52

ABSTRACT

Ceramic products ceramic kiln firing through the complete evaluation of ceramic kiln, advanced, excellent performance and economy of main sintering quality, unit energy consumption, the level of automation, service life, the level of environmental indicators. MCU controlled electric ceramic roller kiln can achieve very good above several technical requirements. The design of heating element for gas valve, evenly arranged in the kiln body, forming a constant temperature zone, the single-chip microcomputer temperature control and fault detection, can effectively improve the quality of products and production efficiency. Realize the use of single-chip roller kiln temperature control, with manual input temperature setting, adjustment parameter setting, heating cycle parameter setting and other functions.

KEYWORDS: Roller kiln temperature control PID setting AT89C52

目 录

1	绪论	1
1.1	课题研究背景	1
1.2	国内研究状况和发展趋势	1
1.3	课题研究内容与目标	2
2	系统组成方案设计	3
2.1	系统方案设计及选择	3
2.2	系统各模块说明	4
3	系统元器件介绍	5
3.1	中央处理器	5
3.1.1	AT89C52 介绍	5
3.1.2	管脚说明	6
3.2	温度传感器	8
3.2.1	温度传感器概述	8
3.2.2	热电偶工作原理	9
3.2.3	热电偶主要特性	10
3.2.4	热电偶的型号	10
3.3	AD574A 模数转换器	11
3.3.1	主要特性	12
3.3.2	AD574A 引脚功能	12
4	辊道窑智能温度控制系统硬件电路设计	14
4.1	前向通道模块设计	14
4.2	温度调节模块电路设计	14
4.3	显示与按键电路设计	15
4.3.1	显示电路设计	15

景德镇陶瓷学院科技艺术学院本科生毕业设计（论文）

4.3.2 按键电路设计	16
4.4 工作状态指示、报警模块设计	18
4.5 温度控制 PID 算法	19
4.5.1 PID 控制器理论	19
4.5.2 PID 控制算法	20
4.6 系统整体电路设计	22
5 软件设计	24
5.1 主程序设计	24
5.2 键盘中断程序流程图设计	36
5.3 温度控制模块程序流程图设计	40
5.4 显示模块程序设计	45
7 总结	46
8 经济分析报告	48
致谢	50
参考文献	51
附录 总电路图	52

1 绪论

1.1 课题研究背景及意义

窑炉智能温度控制系统的目的在于要为窑炉提供更稳定的生产条件和控制精度。在热处理中,通过对生产过程的优化控制和自动工艺管理控制,并且在温度高于陶瓷生产的最大值和最小值时,窑炉发出报警,从而缩短生产周期,提高生产效率和产品质量具有重要的意义。

自 70 年代辊道窑在我国推广应用以来,辊道窑由于窑内温度场均匀、传热速率和效率大、能耗低和烧成周期短等特点,在我国获得了广泛应用。众所周知,温度是陶瓷制品烧成的关键,对辊道窑温度监控的好坏直接关系到陶瓷制品的烧成质量。近年来陶瓷领域的广大科技工作者在这方面作了不懈的努力,设计了众多的辊道窑温度监控系统,总体上可概括为 4 种类型。一是以单片机为核心的智能仪表温度监控系统;二是以工控机为核心的温度监控系统;三是分布式温度监控系统;四是基于现场总线的辊道窑温度监控系统。各自具有自己的优点与不足。

1.2 国内研究状况和发展趋势

早在 20 世纪 80 年代初,我国陶瓷窑炉专家刘振群教授就指出陶瓷窑炉要煤气化、轻体化、辊道化、自动化的四化方向,辊道窑代表了现代陶瓷工业窑炉的发展方向。节能化、稳定化、自动化将是辊道窑的发展趋势。经过我们共同努力,脚踏实地,以理论指导实践,丰富和改进理论,走不断创新之路,相信辊道窑的设计一定能够更上一层楼。

伴随着改革开放的现代化进程,我国陶瓷工业在短短改革开放 30 年的时间内取得了举世瞩目的成就,中国不但成为全球最大的建筑卫生陶瓷生产国,而且是全球最大的建筑卫生陶瓷消费国。辊道窑是近几十年发展起来的新型快烧连续式工业窑炉,具有产量大、能耗低、自动化程度高、劳动强度低、操作方便等诸多优点。近些年来,国内外窑炉设备建造商致力于辊道窑砌筑材料、砌筑技术、窑炉自控技术、燃烧设备装备技术的开发,在提高辊道窑烧成产品质量、产量,降低燃料消耗等各方面上,均取得了很大的成效。

1.3 课题研究内容与目标

本课题设计主要是实现辊道窑的温度智能控制，内容主要包括硬件设计，软件设计，以及调试。总体方案是以 AT89C52 为 CPU 来实现控制系统的运行，主要模块设计包括前向通道温度测量模块，温度调节模块，显示与按键模块，工作状态显示、报警模块等。

研究内容：主要包括个模块的元器件选择，硬件设计与软件设计来实现温度、PID 参数、时间周期等参数的人工输入设定，以作控制用。本设计采用单片机 AT89C52 为控制中心，通过电热偶采集温度信号并将温度信号转化为电量输出，由 AD574A 实现模拟信号转化。温度控制的实现由步进电机调节燃气供给调节阀来实现，同时外接可编程键盘和显示芯片接口 MAX7219，通过小键盘来实现温度值的切换显示，可以通过小键盘设定温度最大与最小值，对于超过或低于最大、最小值进行报警。

研究目的：该系统采用单片机作为控制器采用 PID 算法，温度偏差过大时，系统会通过报警器产生报警信号通知操作人员及时排除故障。窑炉温度时钟处于最佳状态，减少耗能，缩短生产周期，提高产品质量，节约能源。实现辊道窑的温度自适应优化控制，实现温控自动化。

2 系统方案设计

2.1 系统方案设计及选择

方案一：辊道窑控制系统由以下几个部分组成：窑体、单片机系统、电加热元件、热电偶测温元件、阀门驱动系统，如图 2.1 所示。

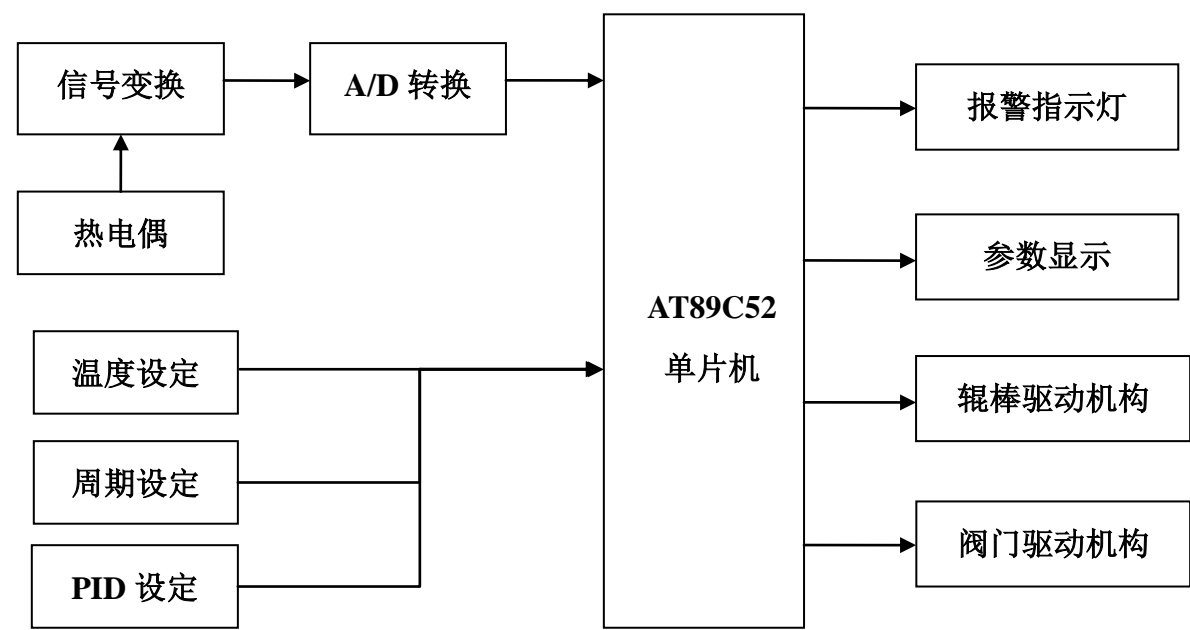


图 2.1 辊道窑系统结构图

控制系统的主要功能就是根据烧成需要自动调节窑体内温度，定时驱动辊棒机构，对于碳化硅棒中电流过大、三相负载不平衡老化以及热电偶断偶等故障，及时发出报警指示信号。单片机系统由 AT89C52 系列单片机组成，采用 12 位模数转换芯片，所有输入输出信号都通过光电隔离芯片进行隔离；单片机控制系统完成温度、时间等参数的设定，以及窑体门、 辊棒驱动机构的开启关闭。

方案二：用可编程控制器作为控制芯片，系统包括显示电路、按键电路、信号采集电路、报警电路、有热电偶检测窑炉内温度，把信号送入 PLC 的转换模块，得到过程量，将此值与给定温度进行比较，然后进行有关运算，得到控制量，最后有控制量控制 PLC 输出的通断。经继电器控制窑炉的工作状态。此方案虽可完成温度控制，但可编程控制器价格昂贵，应用成本太高，性价比低。

考虑性价比因素，我们选择方案一。

2.2 系统各模块说明

（1）参数键盘输入

实现温度、PID 参数、时间周期等参数的人工输入设定，以作控制用。温度设定考虑现场实际情况，不同工艺要求的加热温度各不相同，可以通过键盘进行设定，控制系统自动加热到设定温度；参数设定通过键盘设定温度控制的 PID 调节参数；时间周期设定通过键盘设定加热间隔、辊棒运动时间。

（2）温度检测

采用铂铑—铂热电偶或者铬铂—铑热电偶作为温度传感元件；温度控制采用单点控制，即要求恒温带维持在 1300℃ 预热及降温带根据要求检测。

（3）执行机构

包括电加热用阀门驱动机构。将加热阀门平均分成三组，由于加热时通过调节阀门开度加燃料进行发热，此时单片机输出的控制信号由步进电机完成对阀门开度的控制。辊道窑的机械运动都是通过继电器控制电机使不同的电动连杆机构动作完成的，包括窑体门的开启及辊棒机构运动的控制。

（4）参数、状态指示及故障报警

系统开始运行时，首先查询是否有参数设定键按下判断进入设定或者加热状态；运行过程中，将各个工艺参数显示在数码管上。参数显示表现加热过程中温度、时间等参数。随着温度的增加，其阻值也随之增大到设定值，出现加热元件参数变化过大，影响窑体加热功率，以及温度传感元件老化、断偶等现象，直接影响系统的正常工作。控制系统通过检测有关参数，发现过流、断偶、碳化硅老化或者检测执行机构失灵等故障时，及时发出声光指示信号提醒操作人员，保证系统运行可靠。

3 系统元器件介绍

3.1 中央处理器

3.1.1 AT89C52 介绍

AT89C52 是一个低电压，高性能 CMOS 8 位单片机，片内含 8k bytes 的可反复擦写的 Flash 只读程序存储器和 256 bytes 的随机存取数据存储器（RAM），器件采用 ATMEL 公司的高密度、非易失性存储技术生产，兼容标准 MCS-51 指令系统，片内置通用 8 位中央处理器和 Flash 存储单元，功能强大的 AT89C52 单片机可为您提供许多较复杂系统控制应用场合。

本系统选用 ATMEL 公司生产的 AT89 系列单片机中的 AT89C52。AT89C52 是一个低电压，高性能 CMOS 8 位单片机，片内含 8k bytes 的可反复擦写的 Flash 只读程序存储器和 256 bytes 的随机存取数据存储器（RAM），片内置通用 8 位中央处理器和 Flash 存储单元，功能强大的 AT89C52 单片机可为您提供许多较复杂系统控制应用场合，AT89C52 的主要特点有：

(1)内部程序存储器为电擦除可编程只读存储器 EEPROM，容量 8KB，内部数据存储器容量 256 字节，最大寻址空间 64KB；

(2)AT89C52 有 40 个引脚；三个 16 位定时/计数器；

(3)可利用两根 I/O 口线作为全双工的串行口，有四种工作方式，可通过编程设定；

(4)内部 ROM 中开辟了四个通用工作寄存器区，共 32 个通用寄存器，以适应多种中断或子程序嵌套的情况；

(5)6 个中断源分为两个中断优先级，每个中断源优先级都是可编程的；

(6)内部有一个由直接可位寻址组成的布尔处理机，在指令系统中包含了一个指令子集，专门用于对布尔处理机的各位进行各种布尔处理，特别适用于控制目的和解决逻辑问题；

(7)AT89C52 的状态周期由晶体振荡器 2 分频后获得，作为芯片工作的基本时间单位，在采用 12MHz 晶振时，AT89C52 的状态周期为 $(2/12) = 167\text{ns}$ 。

3.1.2 管脚说明

AT89C52 有 40 个引脚，32 个外部双向输入/输出（I/O）端口，同时内含 2 个外中断口，3 个 16 位可编程定时计数器,2 个全双工串行通信口，2 个读写口线，AT89C52 可以按照常规方法进行编程,但不可以在线编程(S 系列的才支持在线编程)。其将通用的微处理器和 Flash 存储器结合在一起，特别是可反复擦写的 Flash 存储器可有效地降低开发成本。其引脚图如图 3.1 所示。

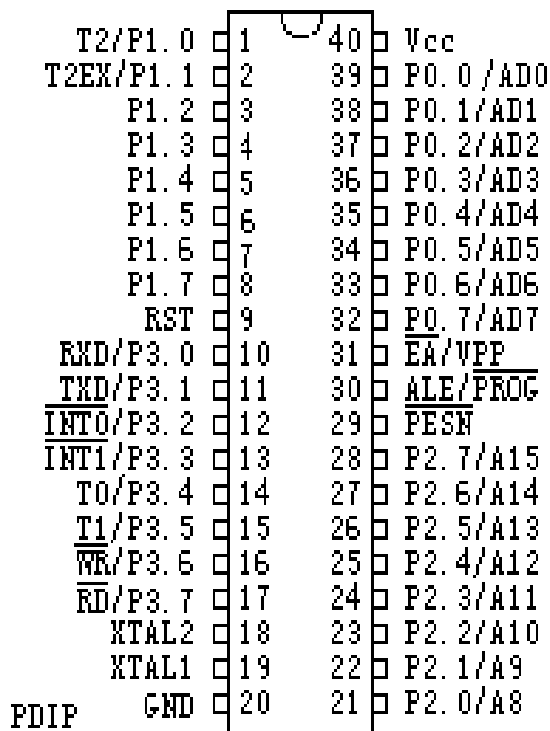


图 3.1 AT89C52 引脚图

VCC: 供电电压。

GND: 接地。

P0 口: P0 口为一个 8 位漏级开路双向 I/O 口，每脚可吸收 8TTL 门电流。当 P0 口的管脚第一次写 1 时，被定义为高阻输入。P0 能够用于外部程序数据存储器，它可以被定义为数据/地址的低八位。在 FLASH 编程时，P0 口作为原码输入口，当 FLASH 进行校验时，P0 输出原码，此时 P0 外部必须接上拉电阻。

P1 口: P1 口是一个内部提供上拉电阻的 8 位双向 I/O 口，P1 口缓冲器能接收输出 4TTL 门电流。P1 口管脚写入 1 后，被内部上拉为高，可用作输入，P1 口被外部下拉为低电平时，将输出电流，这是由于内部上拉的缘故。在 FLASH 编程和校验时，P1 口作

为低八位地址接收。

P2 口：P2 口为一个内部上拉电阻的 8 位双向 I/O 口，P2 口缓冲器可接收，输出 4 个 TTL 门电流，当 P2 口被写“1”时，其管脚被内部上拉电阻拉高，且作为输入。并因此作为输入时，P2 口的管脚被外部拉低，将输出电流。这是由于内部上拉的缘故。P2 口当用于外部程序存储器或 16 位地址外部数据存储器进行存取时，P2 口输出地址的高八位。在给出地址“1”时，它利用内部上拉优势，当对外部八位地址数据存储器进行读写时，P2 口输出其特殊功能寄存器的内容。P2 口在 FLASH 编程和校验时接收高八位地址信号和控制信号。

P3 口：P3 口管脚是 8 个带内部上拉电阻的双向 I/O 口，可接收输出 4 个 TTL 门电流。当 P3 口写入“1”后，它们被内部上拉为高电平，并用作输入。作为输入，由于外部下拉为低电平，P3 口将输出电流（ILL）这是由于上拉的缘故。

P3 口也可作为 AT89C51 的一些特殊功能口，如下表所示：

口管脚 备选功能

P3.0 RXD（串行输入口）

P3.1 TXD（串行输出口）

P3.2 /INT0（外部中断 0）

P3.3 /INT1（外部中断 1）

P3.4 T0（记时器 0 外部输入）

P3.5 T1（记时器 1 外部输入）

P3.6 /WR（外部数据存储器写选通）

P3.7 /RD（外部数据存储器读选通）

P3 口同时为闪烁编程和编程校验接收一些控制信号。

RST：复位输入。当振荡器复位器件时，要保持 RST 脚两个机器周期的高电平时间。

ALE/PROG：当访问外部存储器时，地址锁存允许的输出现电平用于锁存地址的低位字节。在 FLASH 编程期间，此引脚用于输入编程脉冲。在平时，ALE 端以不变的频率周期输出正脉冲信号，此频率为振荡器频率的 1/6。因此它可用作对外部输出的脉冲或用于定时目的。然而要注意的是：每当用作外部数据存储器时，将跳过一个 ALE 脉冲。如想禁止 ALE 的输出可在 SFR8EH 地址上置 0。此时，ALE 只有在执行 MOVX，MOVC

指令是 ALE 才起作用。另外，该引脚被略微拉高。如果微处理器在外部执行状态 ALE 禁止，置位无效。

PSEN: 外部程序存储器的选通信号。在由外部程序存储器取指期间，每个机器周期两次/PSEN 有效。但在访问外部数据存储器时，这两次有效的/PSEN 信号将不出现。

EA/VPP: 当/EA 保持低电平时，则在此期间外部程序存储器（0000H-FFFFH），不管是否有内部程序存储器。注意加密方式 1 时，/EA 将内部锁定为 RESET；当/EA 端保持高电平时，此间内部程序存储器。在 FLASH 编程期间，此引脚也用于施加 12V 编程电源（VPP）。

XTAL1: 反向振荡放大器的输入及内部时钟工作电路的输入。

XTAL2: 来自反向振荡器的输出。

振荡器特性:

XTAL1 和 XTAL2 分别为反向放大器的输入和输出。该反向放大器可以配置为片内振荡器。石晶振荡和陶瓷振荡均可采用。如采用外部时钟源驱动器件，XTAL2 应不接。有余输入至内部时钟信号要通过一个二分频触发器，因此对外部时钟信号的脉宽无任何要求，但必须保证脉冲的高低电平要求的宽度。

3.2 温度传感器

3.2.1 温度传感器概述

本设计中我们采用接触式的温度测量方法，一热电偶作为测温元件，置于工作区中。考虑到测量温度范围和精度，以及价格因素的影响，这里选用装配式热电偶。它具有结构简单、制造方便、精确度高和灵敏度高、信号便于远程传送与集中测量控制等优点，广泛用于 200~1300℃ 范围内的温度。

热电偶式点偶温度计的热敏元件，他测量的基本原理是机遇热点效应（或塞贝克效应）。两种不同的导体将两端相互连接在一起，构成图如图 3.2 所示的闭合回路。

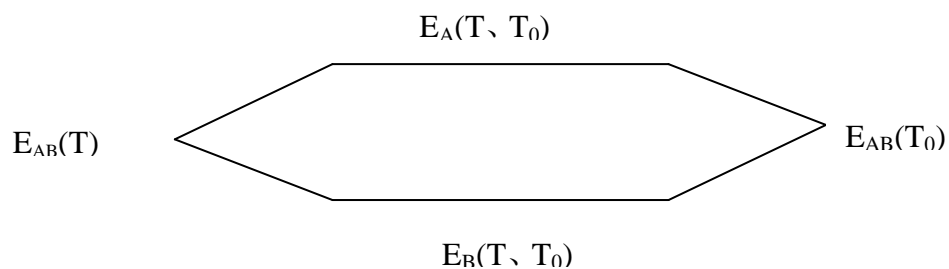


图 3.2 热电偶回路的电势分布

当两个结点的温度不同时，则回路中会产生热电势物理现象称为塞贝克效应（获热电效应）。热这一点是成为塞塞贝克电势，通常有成为热电势，记为 $E_{AB}(T, T_0)$ ，导体 A、B 称为热电极。两热电极 A 和 B 的组合称为热电偶。如图 3.2 左边结点是将两电极焊接在一起，测量时将它放入被测对象中感受、被测温度，故称为测量端获热端；右边结点处于环境之中，要求温度很低，称为参考端获冷端。

热电偶是通过将所测量的温度转换成热电势来实现测温。该热电势实际上由接触电势与温差电视所组成。

3.2.2 热电偶工作原理

热电偶是一种使用最多的一种传感器，他的工作原理室友两种不同的导体或半导体 A 和 B 主城的一个回路，其两端相互连接，只要接点处的温度不一样，一端的温度为 T，另一端的温度为 T_0 ，则回路中就有电流产生，机会路纯在电动势，该电动势成为热电势。

当回路断开时，在断开处 A、B 之间有一电动势 E_T ，其极性和量值与回路中的电动势一致，规定为冷端，当电流由 A 流向 B 是，称 A 为正极，B 为负极。热电势 E_T 与温度差 $(T - T_0)$ 成正比，即

$$E_T = S_{AB}(T - T_0)$$

S_{AB} 为塞贝克系数，又称热电势率，她是热电偶的最重要的特征量，其符号和取决于热电极材料的相对特性。

两种导体的接触电视，不同的金属自由电子密度不同，当两种金属接触在一起时结点处会产生电子扩散，浓度低的得到电子显负电。当扩散大道动态平衡时，得到一个稳定的接触电势。

温度 T 时的接触电势：

$$E_{AB}(T) = \frac{KT}{e} \ln \frac{N_A}{N_B}$$

冷端接触电势

$$E_{AB}(T_0) = \frac{KT_0}{e} \ln \frac{N_A}{N_B}$$

式中：A、B 代表不同的材料。在闭合回路中，总的接触电势为：

$$E_{AB}(T, T_0) = E_{AB}(T) - E_{AB}(T_0) = \frac{K(T - T_0)}{e} \ln \frac{N_A}{N_B}$$

3.2.3 热电偶主要特性

(1) 热电偶可将温度量转换成点亮进行检测，因此对于温度的测量，控制，以及对温度型号的放大、变换都很方便；

(2) 结构简单、制造容易、价格便宜；

(3) 惰性较小、准确度高、测量范围广；

(4) 可做成多种结构，以满足各种测量对象的要求；

(5) 适用于远距离测量与控制。

3.3.4 热电偶的型号

(1) T 型（铜-铜镍）热电偶，铜-铜镍热电偶的测量温区为-200~350℃。

(2) K 型（镍铬-镍硅）热电偶，这种其、热电偶使用温度为-200~1300℃。K 型热电偶具有线性度好，热电动势较大，灵敏度高，稳定性和均匀性较好，抗氧化性能强，价格便宜等优点，能用于氧化性惰性气氛中。广泛为用户所采用。K 型热电偶不能直接在高温下用于硫，还原性或还原，氧化交替的气氛中和真空中，也不推荐用于弱氧化气氛中。

(3) E 型（镍铬-铜镍）热电偶，铬-铜镍热电偶（E 型热电偶）又称镍铬-康铜热电偶，也是一种廉金属的热电偶，该热电偶的使用温度为-200~900℃。E 型热电偶热电动势之大，灵敏度之高属所有热电偶之最，宜制成热电堆，测量微小的温度变化。对于高湿度气氛的腐蚀不甚灵敏，宜用于湿度较高的环境。E 热电偶还具有稳定性好，抗氧化性能优于铜-康铜，铁-康铜热电偶，价格便宜等优点，能用于氧化性和惰性气氛中，广泛为用户采用。

(4) S 型（铂铑 10-铂）热电偶 铂铑 10-铂热电偶俗称单铂铑热电偶。该热电偶长期最高使用温度为 1300℃，短期最高使用温度为 1600℃。S 型热电偶在热电偶系列中具有准确度最高，稳定性最好，测温温区宽，使用寿命长等优点。它的物理，化学性能良好，热电势稳定性及在高温下抗氧化性能好，适用于氧化性和惰性气氛中。S 型热电偶不足之处是热电势，热电势率较小，灵敏读低，高温下机械强度下降，对污染非常敏感，贵金属材料昂贵，因而一次性投资较大。

(5) B 型（铂铑 30-铂铑 6）热电偶 铂铑 30-铂铑 6 热电偶（B 型热电偶）为贵金属热电偶。俗称双铂铑热电偶。该热电偶长期最高使用温度为 1600℃，短期最高使用温度为 1800℃。B 型热电偶在热电偶系列中具有准确度最高，稳定性最好，测温温区宽，使用寿命长，测温上限高等优点。适用于氧化性和惰性气氛中，也可短期用于真空中，但不适用于还原性气氛或含有金属或非金属蒸气气氛中。B 型热电偶一个明显的优点是不需用补偿导线进行补偿，因为在 0~50℃ 范围内热电势小于 3 μV。

因为窑炉的温度一般在 1200℃ 左右要求精度比较高，由以上介绍选用 S 型热电偶作为本设计的温度传感器。

3.3 AD574A 模数转换器

模/数转换器是一种将连续的模拟量转换成离散的数字量的一种电路或者期间。模拟信号转换为数字信号一般需要经过采样保持和量化编码的两个过程，针对不同的采样对象。有不同的 A/D 转换器可供选择，期中有通用的也有专用的。有些 ADC 还包含其他功能，在选择 ADC 器件时需要考虑多种因素，除了关键参数、分辨率和转换速度以外，还需要考虑其他因素，如静态与动态精度、数据接口类型、控制接口与定时、校准能力、通道数量、功耗、使用环境要求、封装形式以及软件有关问题。

根据设计的基本要求，我们选择 AD574A 转换器作为该系统模/数转换器，因为窑炉温度范围是 0~1300℃，分辨率为 $\pm 1^\circ\text{C}$ ，则要求 A/D 转换器的分辨率为 1/1300，一个 12 位的 A/D 转换器分辨率为 $1\text{LSB}=1/4096$ 。若再考虑一些误差，取 12 位为 A/D 转换器是比较适合的，AD574A 的分辨率、精度指标和转换时间都满足系统的要求。所以，我们 AD574A 作为温控系统的模/数转换器。

3.3.1 主要特性

AD574A 是美国模拟数字公司（Analog）推出的单片高速 12 位逐次比较型 A/D 转换器，内置双极性电路构成的混合集成转换显片，具有外接元件少，功耗低，精度高等特点，并且具有自动校零和自动极性转换功能，只需外接少量的阻容件即可构成一个完整的 A/D 转换器，其主要功能特性如下：

- (1)分辨率：12 位
- (2)非线性误差：小于 $\pm 1/2\text{LBS}$ 或 $\pm 1\text{LBS}$
- (3)转换速率：25us
- (4)模拟电压输入范围：0—10V 和 0—20V，0— $\pm 5\text{V}$ 和 0— $\pm 10\text{V}$ 两档四种
- (5)电源电压： $\pm 15\text{V}$ 和 5V
- (6)数据输出格式：12 位/8 位
- (7)芯片工作模式：全速工作模式和单一工作模式

3.3.2 AD574A 引脚功能

AD574A 是单片高速 12 位逐次比较型 A/D 转换器，内部有三态数据数车锁存器，线性误差小于 $1/2\text{LSB}$ ，一次转换时间为 25us，由于芯片内部比较器的输入回路都有可变量程的电阻（ $5\text{K}\Omega$ 或 $10\text{K}\Omega$ ）和双极性偏置电阻，因此 AD574A 的输入模拟信号量程范围有 0—10V 和 0—20V，0— $\pm 5\text{V}$ 和 0— $\pm 10\text{V}$ 两档四种。AD574A 的引脚图如图 3.3 所示：

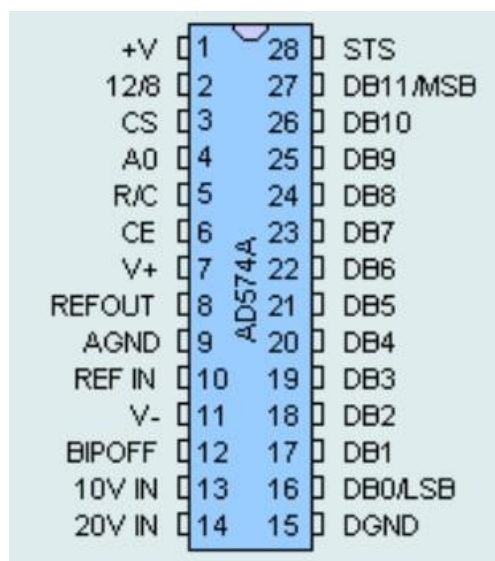


图 3.3 AD574A 的引脚图

引脚功能如下：

- [1]. Pin1(+V)——+5V 电源输入端。
- [2]. Pin2(12/B)——数据模式选择端，通过此引脚可选择数据总线是 12 位或 8 位输出。
- [3]. Pin3(CS)——片选端。
- [4]. Pin4(A0)——字节地址短周期控制端。与 端用来控制启动转换的方式和数据输出格式。须注意的是， 端 TTL 电平不能直接+5V 或 0V 连接。
- [5]. Pin5(R/C)——读转换数据控制端。
- [6]. Pin6(CE)——使能端。
- [7]. Pin7(V+)——正电源输入端，输入+15V 电源。
- [8]. Pin8(REF OUT)——10V 基准电源电压输出端。
- [9]. Pin9(AGND)——模拟地端。
- [10]. Pin10(REF IN)——基准电源电压输入端。
- [11]. Pin(V-)——负电源输入端，输入-15V 电源。
- [12]. Pin1(V+)——正电源输入端，输入+15V 电源。
- [13]. Pin13(10V IN)——10V 量程模拟电压输入端。
- [14]. Pin14(20V IN)——20V 量程模拟电压输入端。
- [15]. Pin15(DGND)——数字地端。
- [16]. Pin16—Pin27(DB0—DB11)——12 条数据总线。通过这 12 条数据总线向外输出 A/D 转换数据。
- [17]. Pin28(STS)——工作状态指示信号端，当 STS=1 时，表示转换器正处于转换状态，当 STS=0 时，声明 A/D 转换结束，通过此信号可以判别 A/D 转换器的工作状态，作为单片机的中断或查询信号之用。

4 辊道窑智能温度控制系统硬件电路设计

4.1 前向通道模块设计

前向通道及温度测量模块，该硬件电路有热电偶、放大器、A/D 转换器组成，该硬件连接如图 4.1 所示。

热电偶测量温度的微弱信号送入控制仪表，经过放大、A/D 转换，然后与给定的值惊醒比较，得到一个偏差值后，按照预先设置 PID 算法进行运算，运算结果经过 D/A 转换，放大转化为 4~20mA 标准的控制信号传送给步进电机执行器，电机执行器根据信号大小控制阀门的开度，步进电机和阀门共同控制燃烧火焰大小来达到控制温度的目的。

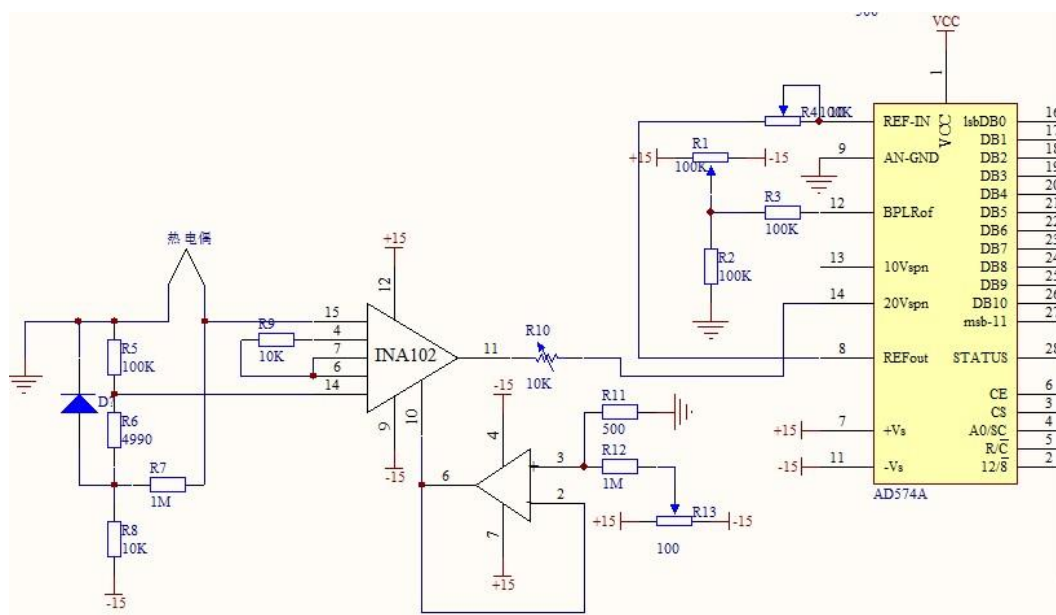


图 4.1 前向通路测量模块电路

4.2 温度调节模块电路设计

步进电机是将脉冲信号转变为角位移或线位移的开环控制元件。在非超载的情况下，电机的转速、停止的位置只取决于脉冲信号的频率与脉冲数，而不受负载的变化影响，即给电机加一个脉冲信号，点击则转过一个步距角。这一线性关系的存在，加上步进电机只有周期性的误差而无积累误差等特点，步进电机被广泛运用于数模转换、速度控制

和位置转换控制系统。

本系统是通过步进电机控制阀门的开度来调节燃气的供给量，从而达到调节窑炉内温度的目的。选用的步进电机的驱动电压为 15V，步进角为 7.5 度，一圈 360 度，需要 48 个脉冲完成，它有 6 根引线序列为：1：红色 2：红色 3：橙色 4：棕色 5：黄色 6：黑色。本系统用 ULN2003 驱动步进电机，则温度调节模块电路如图 4.2 所示

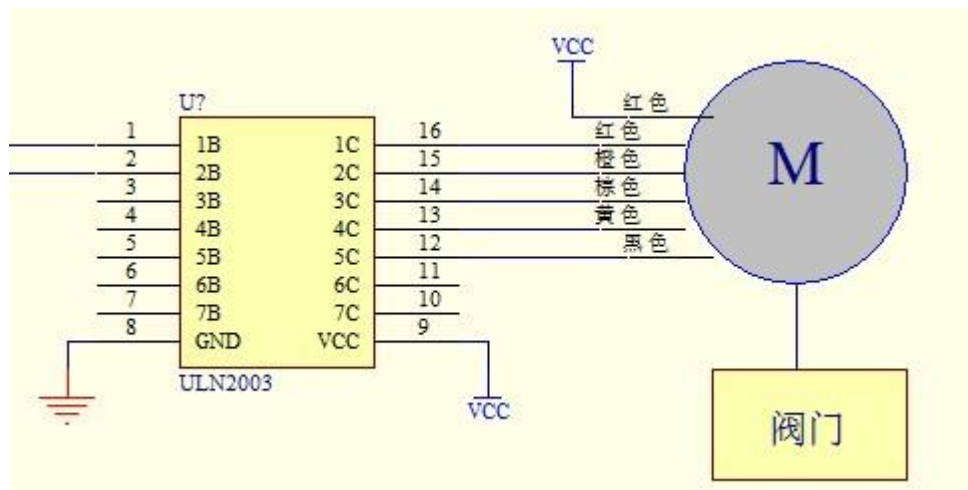


图 4.2 温度调节模块电路

4.3 显示与按键电路设计

4.3.1 显示电路设计

MAX7219 允许一个外部电阻控制显示亮度，外部电阻接于电源输入+5V 和断电流 Iset 端。源于段驱动器的峰值电流，以电流的 100 倍进入 Iset 端。一般段电流为 37mA，允许最大段电流为 40mA，由于 LED 有 2.5V 的电压降，则调节亮度电阻为 2.5V，七段码全部点亮的总电流为 $7 \times 37\text{mA} = 259\text{mA}$ ，外部调节亮度电阻 Rset 最小值时 $2.5\text{V}/259\text{mA} = 9.53\text{K}\Omega$ 。

在 MAX7219 的电源与地并接一个 0.1uF，去耦电容和一个 10uF/16V 电解电容，可以提高其工作可靠性。

在强干扰环境下，干扰源可能通过供电电源获 3 根信号线串入显示电路而造成显示不稳定，从而出现跳跃、显示不全、甚至会全暗或全亮的现象。为此，可通过在 3 根信号线上对地接入一个 1000pF 的瓷片电容来有效的滤除因空间干扰而引起的尖脉冲。显示电路如图 4.3 所示。

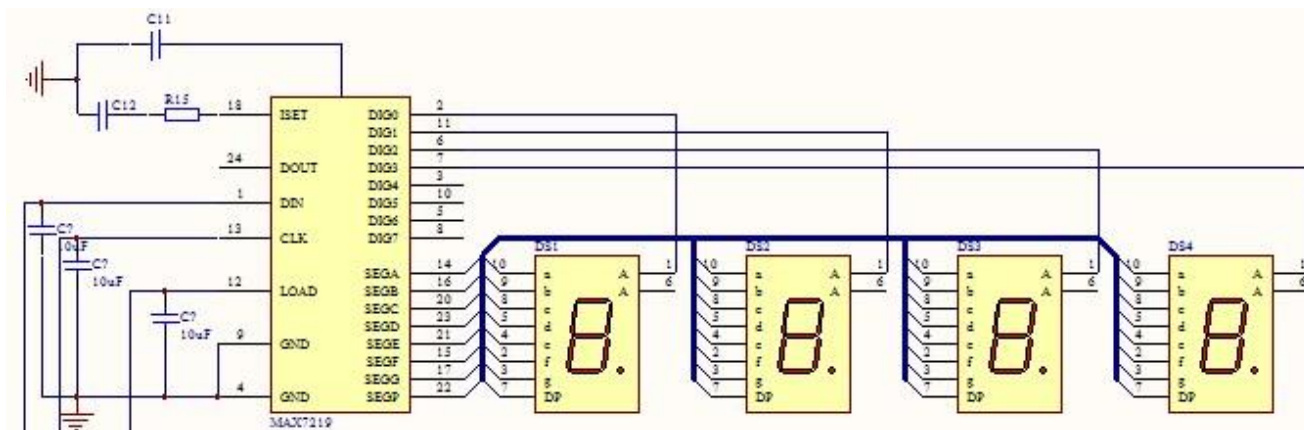


图 4.3 显示电路

4.3.2 按键电路设计

在微机系统中键盘是最常见的输入设备，通常由数字键和功能键组成，其规模取决于系统的要求。

键盘可以分成编码键盘和非编码键盘，编码键盘的按键识别、去抖动、键编码都由硬件完成；非编码键的上述功能在少量的硬件支持下由软件完成。由此可见编码键盘产生键编码的速度快且基本不占 CPU 的时间，但硬件开销大，电路复杂，成本高；非编码键盘则硬件电路简单，成本低，但占用 CPU 的时间长。

键盘接口电路有两个基本特点：

1. 随机性：系统操作人员对键盘操作是随机的，所以操作的键也是随机的；
2. 抖动性：这是由键盘的机械性决定的。

根据这两个特点可以得出以下的接口设计原则：键盘的电平与系统总线电平兼容；单片机能够有效地抑制键盘抖动。消除抖动是由软件实现的，一般采用多数为主的原则；单片机系统能实现对键盘的有效控制，单片机系统键盘接口的目的是为了控制键盘，而键盘电路不能影响总线。

（1）消抖措施：

在一般的电路设计中，按键按下闭合后，应产生一个负脉冲。但是由于在按键按动的同时总有一些抖动，因此在负脉冲的开始和末尾部分总要出现一些齿波，其长短与开关的机械特性有关，一般为 5~10ms。除了抖动之外还有重建，及一个键按下之后接着又按下一键，或者两件同时按下，这些需要采取一定的措施加以消除。

目前消抖的方法有两种，一种是用硬件电路来实现，即用 RC 滤波电路滤除抖动。

另一种是软件延时的方法来解决。在本设计中主要是以软件消抖。主要通过延时来等待信号稳定，在信号稳定后查询键码。其过程是在查询到有按键按下后延时一段时间（12ms~20ms），再查询一尺是否有按键按下，若这一次查询到有键按下，则说明信号已经稳定，然后判断短闭和按键的键码。当闭合按键的键码确定后，再去查询按键是否释放，待按键释放后再进行处理，这样即可消除释放抖动的干扰。重建则以后一次查询为最后结果。

（2） 键盘接口及扫描方式说明：

通过对设计要求的具体分析，在这里采用矩阵式键盘来控制系统参数的输入和调整。矩阵式键盘又称行列式键盘。假设 0 键按下，称为被按键或闭合键，这时矩阵中 A 点行线和列线相通。

行扫描的基本原理：是一条列线为低电平，如果这条线上没有闭合键，则各行线的状态都为高电平；如果列线上有闭合键，则相应的那条行线即变为低电平，这样，就可以根据行线和列线交汇求的闭合键的键码。

行扫描的过程：先使输出口输出 FEH，然后输出行线状态，判断线状态中是否有低电平，如果没有低电平，则使输出口输出 PDH，再判断行线状态，到输出口输出 FCH 时，行线中状态为低电平，则有闭合键。至此，行扫描可以结束，但实际上扫描会继续进行下去，一排除可能出现的多键同时被按下的现象。

键盘中有 4 根行线和 4 根列线，经限流电阻接+5V 电源上，按键跨接在行线和列线上，4×4 行列结构可构成 16 个按键。当无键按下时，74922 芯片的 X、Y 接口处于开路状态。当有按键按下时，与闭合键相连的两条 I/O 口线之间短路。判断有无键按下的方法是：第一步是置 74922 的 X1、X2、X3、X4 为输入状态，从行线输出低电平，读入列线数据，则该列线上有键按下。第二步是置 74922 的 Y1、Y2、Y3、Y4 口为输入状态，从列线输出低电平，读入行线数据，若某一行线为低电平，则该行线上有按键按下。综合一二步的结果，可确定按键的编码号。但是键闭合一次只能进行一次键功能任务，因此须等待按键释放后，再进行键功能造作。通过循环扫描方式可以重复扫描是否有键按下。并在键按下后等待一定时间，在这段时间可以消除按键的抖动。图 4.4 为按键接口电路图。

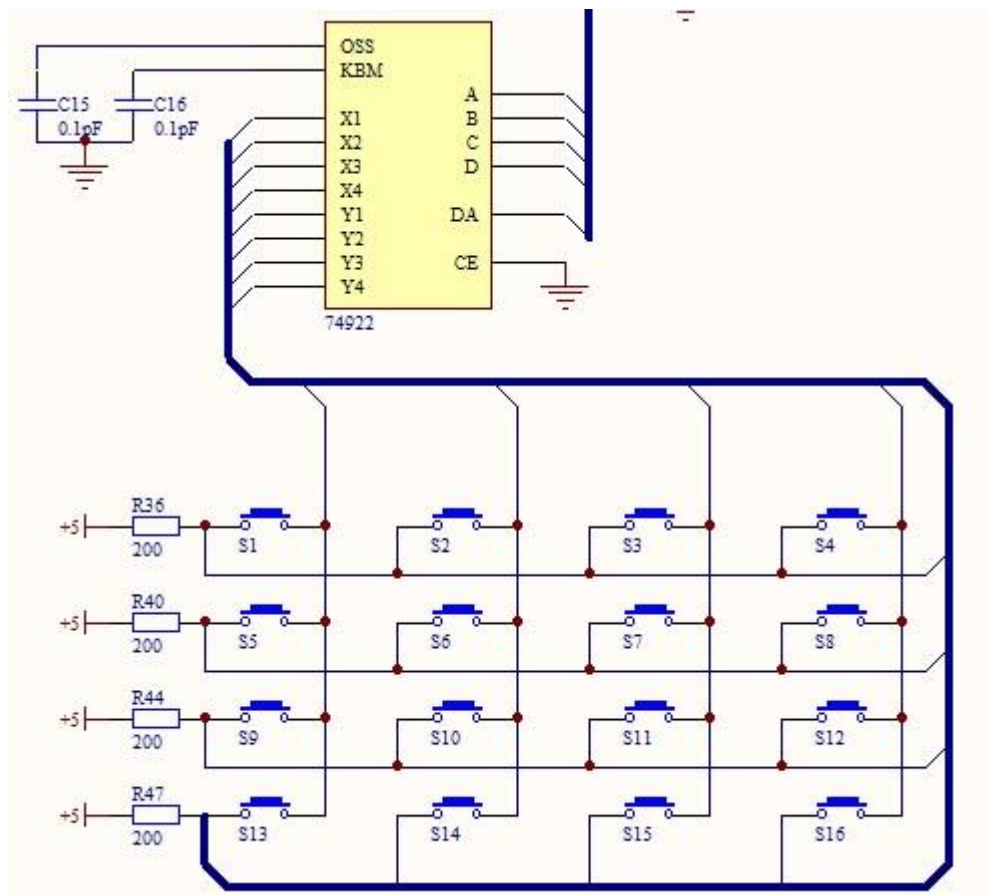


图 4.4 键盘电路

当按下温度按钮时，即可以对窑炉的温度进行设置，此时可以使用数字键设置窑炉需要设置的温度，系统会通过 PID 算法调节控制阀开度以控制系统温度。按下周期按钮是，即可以对窑炉燃烧周期进行设置，此时可以使用数字键设置窑炉燃烧的周期。按下 PID 键，此时可以使用数字键设置窑炉所需的 PID 参数值。当按下手动按钮是，此时不可以通过键盘设置窑炉内的各类参数，但系统会显示窑炉内的内部温度。

4.4 工作状态指示、报警模块设计

工作状态指示、报警模块连线图如图 4.所示，非手动指示灯链接到单片机 P1.0 口，手动指示灯连接到单片机的 P1.1，报警指示灯连接到 P1.2 报警蜂鸣的输出信号通过放大器 Q1 从单片机 P1.3 口输出。

在仪表上设定的超温保护一般比高温区温控点的正常工作温度搞 20~30 ℃，当窑炉内温度超过设定的保护温度，电铃及信号灯会发出声光报警信号。系统设置了 3 个指示灯，一个为报警指示灯，一个为周期设置指示灯，另外一个为 PID 参数设置指示灯。当

温控系统出现故障时报警指示灯亮，当按下周期按键时，周期设置指示灯亮，当 PID 参数按键按下，PID 参数设置指示灯亮。

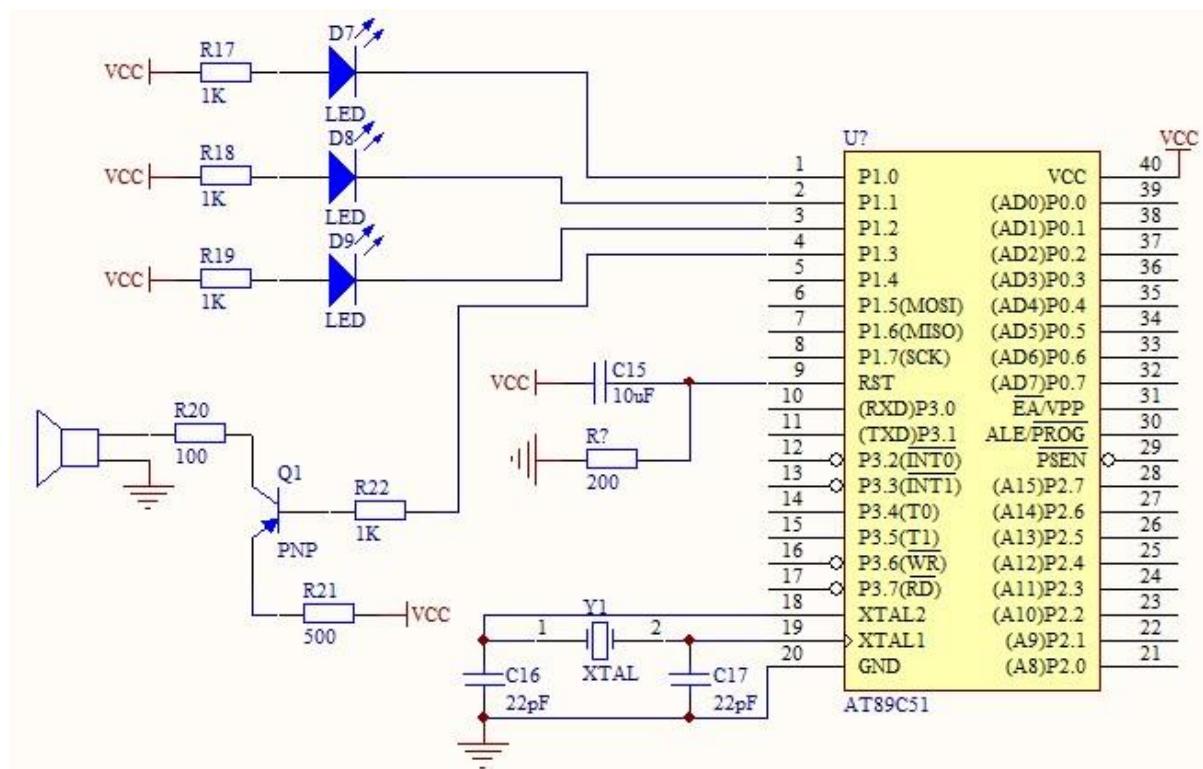


图 4.6 报警模块电路

4.5 温度控制 PID 算法

4.5.1 PID 控制器理论

PID 控制器是一种线性控制器，它根据给定值 $r(t)$ 与实际输出值 $y(t)$ 构成控制偏差 $e(t)$ ：

$$e(t) = r(t) - y(t) \quad (4-1)$$

将偏差 $e(t)$ 的比例(Proportional)、积分(Integral)和微分(Derivative)通过线性组合构成控制量，对被控对象进行控制，因此称为 PID 控制，PID 控制系统原理如图 4.7 所示：

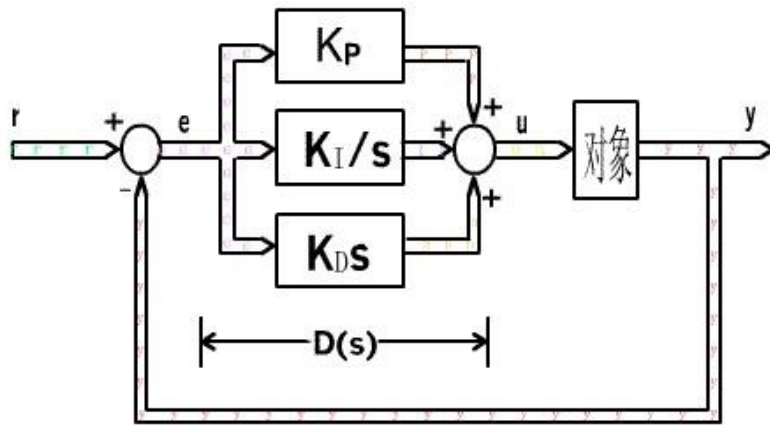


图 4.7 PID 控制系统原理图

其控制规律为

$$u(t) = K_p[e(t) + \frac{1}{T} \int_0^t e(t)dt + T_D \frac{de(t)}{dt}] \quad (4-2)$$

或者写成传递函数形式为

$$G(s) = K_p(1 + \frac{1}{T_{is}} + T_{Ds}) \quad (4-3)$$

式 4-3 中 K_p ：比例系数； K_I ：积分时间常数； K_D ：微分时间常数。

PID 控制器各校正环节的作用如下：

- （1）比例环节即时成比例地反映控制系统的偏差信号 $e(t)$ ，偏差一旦产生，控制器立即产生控制作用，以减少偏差；
- （2）积分环节主要用于消除静差，提高系统的无差度；
- （3）微分环节能够反映偏差信号的变化趋势(变化速率)，并且能在偏差信号值变得太大之前，在系统中引入一个有效的早期修正信号，从而加快系统的动作速度，减少调节时间。

4.5.2 PID 控制算法

由于计算机控制是一种采样控制系统，它只能根据采样时刻的偏差值计算控制量。因此，式 4-3 中的积分和微分项不能直接使用，需要进行离散化处理。现令 T 为采样周期，以一系列的采样时刻点 KT 代表连续时间 t ，以累加求和近似代替积分以一阶后向差分近

似代替微分做如下的近似变换：

$$t = KT \quad (4-4)$$

$$\int_0^t e(t) \approx T \sum_{j=0}^k e(jT) = T \sum_{j=0}^k e(j) \quad (4-5)$$

$$\frac{de(t)}{dt} \approx \frac{e(KT) - e[(k-1)T]}{T} = \frac{e(k) - e(k-1)}{T} \quad (4-6)$$

其中， T 为采样周期， $e(k)$ 为系统第 k 次采样时刻的偏差值， $e(k-1)$ 为系统第 $(k-1)$ 次采样时刻的偏差值， k 为采样序号， $k=0, 1, 2, \dots$ 。

将上面的式 4-4 和式 4-5 代入式 4-6 则可以得到离散的 PID 表达式：

$$u(k) = K_p \{e(k) + \frac{T}{T_i} \sum_{j=0}^k e(j) + \frac{T_d}{T} [e(k) - e(k-1)]\} \quad (4-7)$$

如果采样周期了足够小，该算式可以很好的逼近模拟 PID 算式，因而使被控过程与连续控制过程十分接近。通常把式 4-7 称为 PID 的位置式控制算法。

若在式 4-7 中，令：

$$K_I = \frac{K_p T}{T_i} \quad (\text{称为积分系数})$$

$$K_D = \frac{K_p T_d}{T} \quad (\text{称为微分系数})$$

则

$$u(k) = K_I e(k) + K_I \sum_{j=0}^k e(j) + K_D [e(k) - e(k-1)] \quad (4-8)$$

(4-8)式即为离散化的位置式 PID 控制算法的编程表达式。可以看出，每次输出与过去的所有状态都有关，要想计算 $u(k)$ ，不仅涉及 $e(k)$ 和 $e(k-1)$ ，且须将历次 $e(j)$ 相加，计算复杂，浪费内存。下面，推导计算较为简单的递推算式。为此，对(4-8)式作如下的变动：

考虑到第 $(k-1)$ 次采样时有：

$$u(k-1) = K_p \{e(k-1) + \frac{T}{T_i} \sum_{j=0}^{k-1} e(j) + \frac{T_d}{T} [e(k-1) - e(k-2)]\} \quad (4-9)$$

使 (4-8) 两边对应减去 (4-9) 式得

$$u(k) - u(k-1) = K_p \{ e(k) - e(k-1) + \frac{T}{T_I} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)] \}$$

整理后得

$$u(k) = u(k-1) + a_0 e(k) - a_1 e(k-1) + a_2 e(k-2) \quad (4-10)$$

$$\text{其中: } a_0 = K_p \left(1 + \frac{T}{T_I} + \frac{T_D}{T}\right), \quad a_1 = K_p \left(1 + \frac{2T_D}{T}\right), \quad a_2 = K_p \frac{T_D}{T}$$

式（4-10）就是 PID 位置式的递推形式

如果令 $\Delta u(k) = u(k) - u(k-1)$ ，则：

$$\Delta u(k) = a_0 e(k) - a_1 e(k-1) + a_2 e(k-2) \quad \text{式（4-11）}$$

式中 a_0 、 a_1 、 a_2 同式（4-10）中一样。

因为在计算机控制中式中 a_0 、 a_1 、 a_2 都可以事先求出，所以，实际控制时只须获得 $e(k)$ 、 $e(k-1)$ 、 $e(k-2)$ 三个有限的偏差值就可以求出控制增量。由于其控制输出对应执行机构的位置的增量，故(4-11)式通常被称为 PID 控制的增量式算式^[3]。增量式 PID 控制算法与位置式控制算法比较，有如下的一些优点：

（1）位置式算法每次输出与整个过去状态有关，算式中要用到过去偏差的累加值 $\sum e(j)$ ，容易产生较大的累计误差。而增量式中只须计算增量，控制增量的确定仅与最近几次偏差采样值有关，当存在计算误差或者精度不足时，对控制量的影响较小，且较容易通过加权处理获得比较好的控制效果；

（2）由于计算机只输出控制增量，所以误动作影响小，而且必要时可以用逻辑判断的方法去掉，对系统安全运行有利；

（3）手动与自动切换时冲击比较小。

4.6 系统整体电路设计

系统总体电路主要有前向通道电路、温度调节电路、显示与键盘电路、工作状态指示与报警模块电路四部分组成。前向通道模块电路主要完成温度信号采集与转化可供单片机分析；温度调节主要是通过 PID 算法对阀门进行开度进行调节实现温度控制；显示

与键盘电路主要有由四个数码管组成的显示盘与 4*4 的矩阵输入键盘组成，可实现温度的实时显示，以及窑炉各参数的设定；工作状态指示与报警模块电路主要由蜂鸣器与发光二极管组成，实现报警与工作指示的功能。

系统整体电路图设计见附录。

5 软件设计

5.1 主程序设计

本设计主程序流程图如图 5.1 所示。

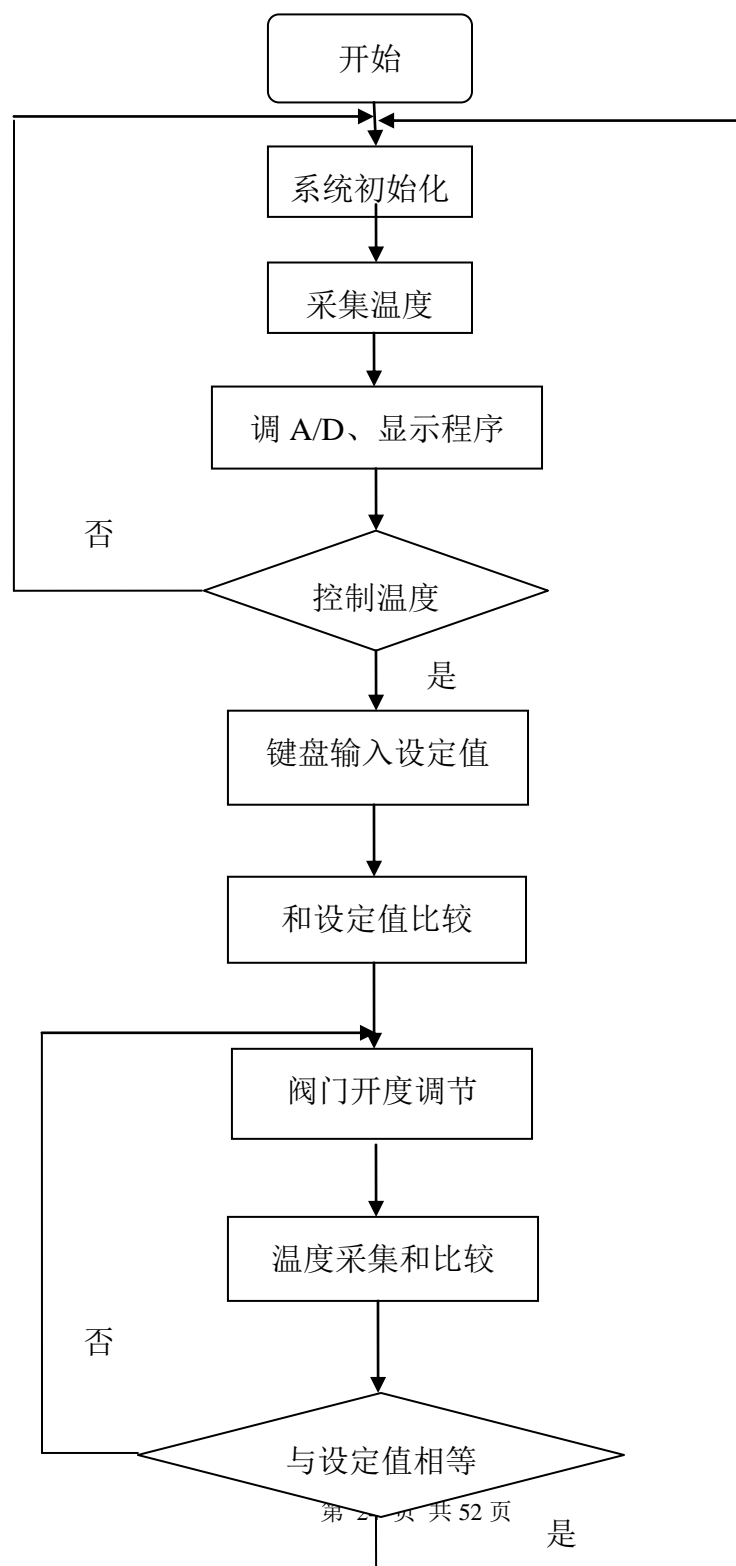


图 6.1 主程序流程图

程序启动后，嗽跄清理系统内存，然后进行信号采集，通过 A/D 转换后，信号传送到单片机，由单片机控制显示设备，显示实时温度，然后进入待机状态，等待键盘输入温度，周期，PID 参数，然后将设定的温度参数鱼现在的温度进行比较，得出结果后，启动电机调节阀门开度调节温度。

主程序如下所示

```
#include<reg52.h>
#include<intrins.h>
#include<math.h>
#include<string.h>

struct PID {
    unsigned int SetPoint; // 设定目标 Desired Value
    unsigned int Proportion; // 比例常数 Proportional Const
    unsigned int Integral; // 积分常数 Integral Const
    unsigned int Derivative; // 微分常数 Derivative Const
    unsigned int LastError; // Error[-1]
    unsigned int PrevError; // Error[-2]
    unsigned int SumError; // Sums of Errors
};

struct PID spid; // PID Control Structure
unsigned int rout; // PID Response (Output)
unsigned int rin; // PID Feedback (Input)

sbit data1=P1^0;
sbit clk=P1^1;
sbit plus=P2^0;
sbit subs=P2^1;
sbit stop=P2^2;
sbit output=P3^4;
```

```
sbit DQ=P3^3;
sbit d=P3^1;
sbit cp=P3^2;
sbit dir=P3^5;
unsigned char flag,flag_1=0;
unsigned char high_time,low_time,count=0;// 占空比调节参数
unsigned char set_temper=35;
unsigned char temper;
unsigned char i;
unsigned char j=0;
unsigned int s;
/*****
延时子程序,延时时间以 12M 晶振为准,延时时间为 30us×time
*****/
void delay(unsigned char time)
{
unsigned char m,n;
for(n=0;n<time;n++)
for(m=0;m<2;m++){ }
}
/*****
写一位数据子程序
*****/
void write_bit(unsigned char bitval)
{
EA=0;
DQ=0; /*拉低 DQ 以开始一个写时序*/
if(bitval==1)
```

```
{
_nop_();
DQ=1; /*如要写 1，则将总线置高*/
}

delay(5); /*延时 90us 供 DA18B20 采样*/
DQ=1; /*释放 DQ 总线*/
_nop_();
_nop_();
EA=1;
}

/*****
写一字节数据子程序
*****/

void write_byte(unsigned char val)
{
unsigned char i;
unsigned char temp;
EA=0;          /*关中断*/
TR0=0;
for(i=0;i<8;i++) /*写一字节数据，一次写一位*/
{
temp=val>>i; /*移位操作，将本次要写的位移到最低位*/
temp=temp&1;
write_bit(temp); /*向总线写该位*/
}
delay(7); /*延时 120us 后*/
// TR0=1;
EA=1; /*开中断*/
```



```
}

/*****

读一位数据子程序

*****/

unsigned char read_bit()
{
    unsigned char i,value_bit;
    EA=0;
    DQ=0; /*拉低 DQ，开始读时序*/
    _nop_();
    _nop_();
    DQ=1; /*释放总线*/
    for(i=0;i<2;i++){ }
    value_bit=DQ;
    EA=1;
    return(value_bit);
}

/*****

读一字节数据子程序

*****/

unsigned char read_byte()
{
    unsigned char i,value=0;
    EA=0;
    for(i=0;i<8;i++)
    {
        if(read_bit()) /*读一字节数据，一个时序中读一次，并作移位处理*/
            value|=0x01<<i;
    }
}
```

```
delay(4); /*延时 80us 以完成此次都时序，之后再读下一数据*/
}
EA=1;
return(value);
}

/*****

复位子程序

*****/

unsigned char reset()
{
    unsigned char presence;
    EA=0;
    DQ=0; /*拉低 DQ 总线开始复位*/
    delay(30); /*保持低电平 480us*/
    DQ=1; /*释放总线*/
    delay(3);
    presence=DQ; /*获取应答信号*/
    delay(28); /*延时以完成整个时序*/
    EA=1;
    return(presence); /*返回应答信号，有芯片应答返回 0,无芯片则返回 1*/
}

/*****

获取温度子程序

*****/

void get_temper()
{
    unsigned char i,j;
    do
```

```
{
i=reset(); /*复位*/
}while(i!=0); /*1 为无反馈信号*/
i=0xcc; /*发送设备定位命令*/
write_byte(i);
i=0x44; /*发送开始转换命令*/
write_byte(i);
delay(180); /*延时*/
do
{
i=reset(); /*复位*/
}while(i!=0);
i=0xcc; /*设备定位*/
write_byte(i);
i=0xbe; /*读出缓冲区内容*/
write_byte(i);
j=read_byte();
i=read_byte();
i=(i<<4)&0x7f;
s=(unsigned int)(j&0x0f);
s=(s*100)/16;
j=j>>4;
temper=i|j; /*获取的温度放在 temper 中*/
}

/*****

Initialize PID Structure

*****/

void PIDInit (struct PID *pp)
```

```
{
memset ( pp,0,sizeof(struct PID));
}

/*****

PID 计算部分

*****/

unsigned int PIDCalc( struct PID *pp, unsigned int NextPoint )
{
unsigned int dError,Error;

Error = pp->SetPoint - NextPoint; // 偏差
pp->SumError += Error; // 积分
dError = pp->LastError - pp->PrevError; // 当前微分
pp->PrevError = pp->LastError;
pp->LastError = Error;
return (pp->Proportion * Error//比例
+ pp->Integral * pp->SumError //积分项
+ pp->Derivative * dError); // 微分项
}

/*****

温度比较处理子程序

*****/

compare_temper()
{
unsigned char i;
if(set_temper>temper)
{
if(set_temper-temper>1)
{
```

```
high_time=100;
low_time=0;
}
else
{
for(i=0;i<10;i++)
{ get_temper();
rin = s; // Read Input
rout = PIDCalc ( &spid,rin ); // Perform PID Iteration
}
if (high_time<=100)
high_time=(unsigned char)(rout/800);
else
high_time=100;
low_time= (100-high_time);
}
}
else if(set_temper<=temper)
{
if(temper-set_temper>0)
{
high_time=0;
low_time=100;
}
else
{
for(i=0;i<10;i++)
{ get_temper();
```

```
rin = s; // Read Input
rout = PIDCalc ( &spid,rin ); // Perform PID Iteration
}
if (high_time<100)
high_time=(unsigned char)(rout/10000);
else
high_time=0;
low_time= (100-high_time);
}
}
// else
// {}
}

/*****
T0 中断服务子程序，用于控制电平的翻转 ,40us*100=4ms 周期
*****/

void serve_T0() interrupt 1 using 1
{
if(++count<=(high_time))
output=1;
else if(count<=100)
{
output=0;
}
else
count=0;
TH0=0x2f;
TL0=0xe0;
```

```
}

/*****

串行口中断服务程序，用于上位机通讯

*****/

void serve_sio() interrupt 4 using 2
{
/* EA=0;

RI=0;

i=SBUF;

if(i==2)
{
while(RI==0){ }
RI=0;

set_temper=SBUF;

SBUF=0x02;

while(TI==0){ }

TI=0;

}

else if(i==3)
{
TI=0;

SBUF=temper;

while(TI==0){ }

TI=0;

}

EA=1; */

}

void disp_1(unsigned char disp_num1[6])
```

```
{
unsigned char n,a,m;
for(n=0;n<6;n++)
{
// k=disp_num1[n];
for(a=0;a<8;a++)
{
clk=0;
m=(disp_num1[n]&1);
disp_num1[n]=disp_num1[n]>>1;
if(m==1)
data1=1;
else
data1=0;
_nop_();
clk=1;
_nop_();
}
}
}
```

/******

显示子程序

功能：将占空比温度转化为单个字符，显示占空比和测得到的温度

*****/

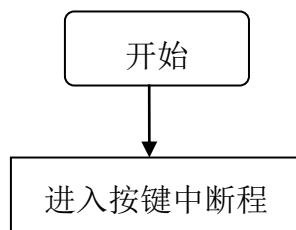
```
void display()
{
unsigned char code number[]={0xfc,0x60,0xda,0xf2,0x66,0xb6,0xbe,0xe0,0xfe,0xf6};
unsigned char disp_num[6];
```



```
unsigned int k,k1;
k=high_time;
k=k%1000;
k1=k/100;
if(k1==0)
disp_num[0]=0;
else
disp_num[0]=0x60;
k=k%100;
disp_num[1]=number[k/10];
disp_num[2]=number[k%10];
k=temper;
k=k%100;
disp_num[3]=number[k/10];
disp_num[4]=number[k%10]+1;
disp_num[5]=number[s/10];
disp_1(disp_num);
}
```

5.2 键盘中断程序流程图设计

按键中断程序主要包括读键和键值两部分，因为中断函数不能返回值，要在中断函数中读出按键的值所以必须通过修改对应的全局变量 SCAN 来传递。其流程图如图 5.2 所示。



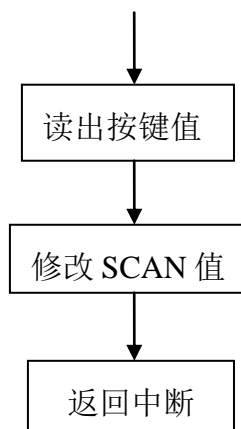


图 5.2 键盘中断流程

键盘中断程序如下：

```
void set_temp()          //键盘调整函数
```

```
{

    if (zengjia==0) pid.set++;
    while(zengjia==0)
        xianshi() ;
    if (jianshao==0) pid.set--;
    while(jianshao==0)
        xianshi()      ;
    if (pid.set<0) pid.set=0;
}
```

```
void pid_con(struct  PID *pid,uint value2)
```

```
{
    int error;
    pid->currerror=pid->set-value2;
```

```
pid->sumerror+=pid->currerror;
error=pid->currerror-pid->lasterror;
result=pid->pgain*(pid->currerror+pid->sumerror*pid->igain+pid->dgain*error )+10
;

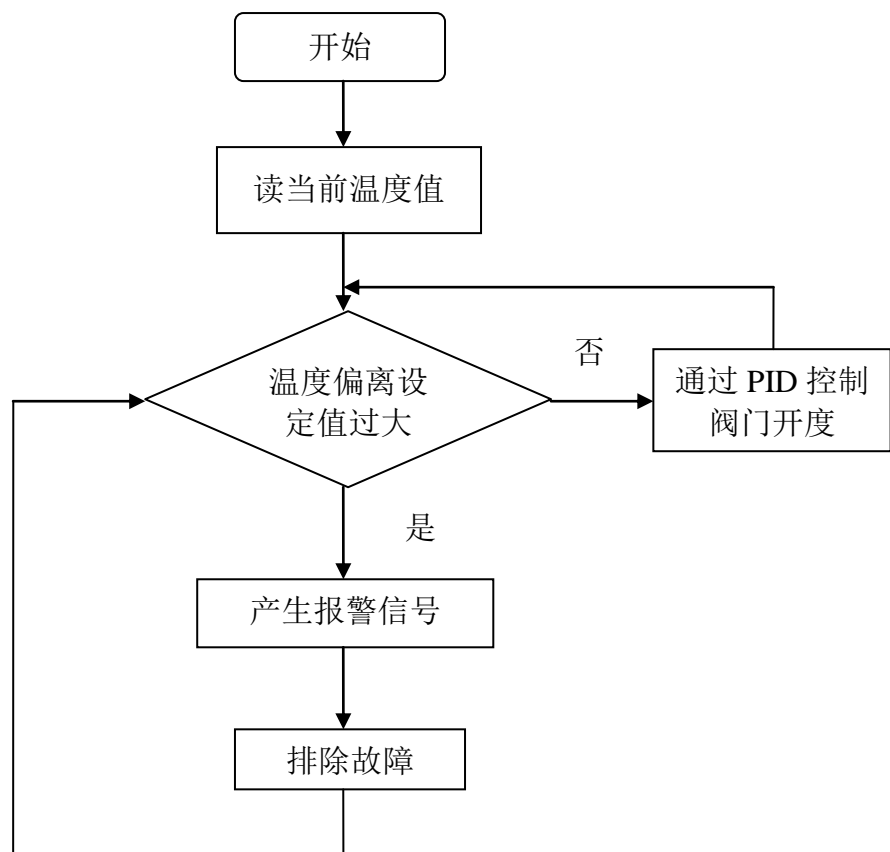
pid->prerror=pid->lasterror;
pid->lasterror=pid->currerror;
}
```

```
void adpid(uint value2,struct PID *pid)
{
    pid_con(pid,value2);

    if(pid->set>value2)
        if(pid->set-value2<=pid->deadband)
        {
            if(result>300)    result=300;
        }
    else
    {
        result=300;
        pid->sumerror=0;
    }
    if(pid->set<=value2 )
    {
        if(result>300)    result=0;
    }
}
```


5.3 温度控制模块程序流程图设计

温度控制模版主要是由 AD574A 转换后的信号，得出实时温度值再与通过键盘设定的温度值进行比较，再通过 PID 算法来调节阀门开度调节温度值，由报警模块进行过温保护。温度控制流程图如图 5.3 所示。



5.3 温度控制流程图

温度控制模块程序如下所示：

```

/*****
获取温度子程序
*****/

void get_temper()
    
```

```
{
unsigned char i,j;
do
{
i=reset();/*复位*/
}while(i!=0);/*1 为无反馈信号*/
i=0xcc; /*发送设备定位命令*/
write_byte(i);
i=0x44; /*发送开始转换命令*/
write_byte(i);
delay(180); /*延时*/
do
{
i=reset();/*复位*/
}while(i!=0);
i=0xcc; /*设备定位*/
write_byte(i);
i=0xbe; /*读出缓冲区内容*/
write_byte(i);
j=read_byte();
i=read_byte();
i=(i<<4)&0x7f;
s=(unsigned int)(j&0x0f);
s=(s*100)/16;
j=j>>4;
temper=i|j; /*获取的温度放在 temper 中*/
}
/*****
```

Initialize PID Structure

```
*****/
```

```
void PIDInit (struct PID *pp)
```

```
{
```

```
memset ( pp,0,sizeof(struct PID));
```

```
}
```

```
*****
```

PID 计算部分

```
*****/
```

```
unsigned int PIDCalc( struct PID *pp, unsigned int NextPoint )
```

```
{
```

```
unsigned int dError,Error;
```

```
Error = pp->SetPoint - NextPoint; // 偏差
```

```
pp->SumError += Error; // 积分
```

```
dError = pp->LastError - pp->PrevError; // 当前微分
```

```
pp->PrevError = pp->LastError;
```

```
pp->LastError = Error;
```

```
return (pp->Proportion * Error//比例
```

```
+ pp->Integral * pp->SumError //积分项
```

```
+ pp->Derivative * dError); // 微分项
```

```
}
```

```
*****
```

温度比较处理子程序

```
*****/
```

```
compare_temper()
```

```
{
```

```
unsigned char i;
```

```
if(set_temper>temper)
```

```
{
if(set_temper-temper>1)
{
high_time=100;
low_time=0;
}
else
{
for(i=0;i<10;i++)
{ get_temper();
rin = s; // Read Input
rout = PIDCalc ( &spid,rin ); // Perform PID Iteration
}
if (high_time<=100)
high_time=(unsigned char)(rout/800);
else
high_time=100;
low_time= (100-high_time);
}
}
else if(set_temper<=temper)
{
if(temper-set_temper>0)
{
high_time=0;
low_time=100;
}
else
```



```
{
for(i=0;i<10;i++)
{ get_temper();
rin = s; // Read Input
rout = PIDCalc ( &spid,rin ); // Perform PID Iteration
}
if (high_time<100)
high_time=(unsigned char)(rout/10000);
else
high_time=0;
low_time= (100-high_time);
}
}
// else
// {}
}

/*****
T0 中断服务子程序，用于控制电平的翻转 ,40us*100=4ms 周期
*****/

void serve_T0() interrupt 1 using 1
{
if(++count<=(high_time))
output=1;
else if(count<=100)
{
output=0;
}
else
```

```
count=0;
TH0=0x2f;
TL0=0xe0;
}
```

5.4 显示模块程序设计

显示程序如下所示

```
/******
```

显示子程序

功能：将占空比温度转化为单个字符，显示占空比和测得到的温度

```
*****/
```

```
void display()
{
    unsigned char code number[]={0xfc,0x60,0xda,0xf2,0x66,0xb6,0xbe,0xe0,0xfe,0xf6};
    unsigned char disp_num[6];
    unsigned int k,k1;
    k=high_time;
    k=k%1000;
    k1=k/100;
    if(k1==0)
        disp_num[0]=0;
    else
        disp_num[0]=0x60;
    k=k%100;
    disp_num[1]=number[k/10];
    disp_num[2]=number[k%10];
    k=temper;
    k=k%100;
```

```
disp_num[3]=number[k/10];  
disp_num[4]=number[k%10]+1;  
disp_num[5]=number[s/10];  
disp_1(disp_num);  
}
```

7 总结

陶瓷产品的烧成主要通过陶瓷窑炉完成，评价陶瓷窑炉的先进性、优良性和经济性主要有烧成质量、单位能耗、自动化水平、使用寿命、环保水平等方面的指标。用单片机控制的电能陶瓷辊道窑能很好达到以上几个技术指标的要求。本设计加热元件为碳化硅棒，均匀地布置于窑体，形成一个 1300 摄氏度的恒温带，由单片机进行温度控制及故障检测，可以有效地提高产品质量和生产效率。

利用单片机实现辊道窑的温度控制，具有人工输入温度设定、调节参数设定、加热循环周期参数设定等功能，还可以解决加热硅碳棒老化所带来的负载不平衡及功率波动问题。

本设计主要是以单片机为核心的智能仪表温度监控系统，具有人工输入温度设定、调节参数设定、加热循环周期参数设定等功能。这种监控系统的特点是单片机智能显示控制仪表本身具有自动校正零漂的功能，加上每支测量热电偶都配有温度补偿导线，使得仪表的测量不受环境温度的影响，从而做到测量准确和稳定。这种监控系统已应用于陶瓷墙地砖气烧辊道窑中，它可使地面砖的 A 级品率达到 80%以上。实验证明，其各项经济技术指标均达到或超过了相关的意大利引进窑炉水平，已被国内数家陶瓷厂采用，均取得明显的经济效益。

用单片机控制的电能陶瓷辊道窑能很好达到以上几个技术指标的要求。本设计加热元件为碳化硅棒，均匀地布置于窑体，形成一个 1300 摄氏度的恒温带，由单片机进行温度控制及故障检测，可以有效地提高产品质量和生产效率。系统组成与工作。

8 经济分析报告

本设计是辊道窑智能温度控制系统，广泛用于中小型陶瓷企业。辊道窑一般截面较小，窑内温度均匀，适合快速烧成，但辊子材质和安装技术要求较高。主要用于建筑卫生陶瓷制品的快速烧成。

电窑靠发热体的辐射对制品加热，因温差大，上下约为 30 摄氏度。而燃气窑炉除辐射传热首位，燃气的高温高速流动加强了对流换热。经计算，燃气出口速度可达 16m/s。因而窑炉内温度均匀，温差小，上下约 10 摄氏度而且温差大于 10 摄氏度会发出报警，保证窑炉内温度稳定性，成本价格 如下表所示。

表 8.1 元器件成本价格表

器件	数量	总价（元）
电阻	45 个	5
滑动变阻	5	4
二极管	3	2
发光二极管	5	2
电容	15	1.5
AD 转换器（AD574A）	1	30
数据选择器 74LS153	1	1.5
石英晶体振荡器	1	1
7805	1	1
7815	1	1
AT89C52	1	10
MAX7219	1	10
UNL2003	1	9
数码管	4	3.6
与非门 74LSC3	1	1
74922	1	2

景德镇陶瓷学院科技艺术学院本科生毕业设计（论文）

PNP	1	5
扬声器	1	1
按键	16	6.4
变压器	2	50
步进电机	2	700

整个系统比较贵的元器件有 AD574A 约 30 元，AT89C52 约 10 元，MAX7219 约 10 元，ULN2003 约 8 元与非门 74LSC3 约 4 元，变压器约 40 元，步进电机 350 元，计算总成本价格约为 847 元。

由于现在提倡环保型社会，本产品相对于其他产品来说，更节省原材料，废气排放量更少，产品质量优良等优点，它有更好的发展前景。而且市场上本产品的市场价格为 1400 元左右，我设计的产品预售价格约为 1100 元，利润为 200 元左右

我国科技运用程度与发达国家相比还有较大差距，通过运用高科技参评可以提高我国科技产品的竞争力。用单片机设计窑炉智能温度控制系统，有经济，低碳等各种优点。

致谢

历时五个月的时间在老师和同学的帮助下将这篇论文写完，四年学习成长的积累浓缩于这短短的篇幅，满怀感激！

感谢我的室友、我的 10 自动化！感谢在陶院匆匆四年光阴带给我改变！

感谢在论文制作过程中给予指导和教诲我专业、严谨的朱永红老师！

感谢在百忙之中评阅论文和参加答辩的评委老师！

谢谢辛勤付出的老师！你们最可爱！

参考文献

1. 沈红卫. 单片机应用系统设计实例与分析. 北京: 北京航空航天大学出版社, 2003
2. 胡汉才. 单片机原理及其接口技术. 北京: 清华大学出版社, 1996
3. 于海生. 微型计算机控制技术. 北京: 清华大学出版社, 1999
4. 张俊谟. 单片机中级教程-原理及应用. 北京: 北京航空航天大学出版社, 2000
5. 王明顺, 吴省. 可涓流充电的串行实时时钟芯片 DS1302 及其应用设计. 电子技术应用, 1996(10)
6. 梁森, 欧阳三泰. 自动检测技术及应用. 机械工程出版社, 2008
7. 和卫星等. 电子电路 CAD 实用技术. 中国科学技术大学出版社, 2001
8. 乐建波. 温度控制系统. 化学工业出版社, 2007
9. 康华光, 邹寿彬. 电子技术基础. 北京: 高等教育出版社, 2000
10. 吴振顺. 自适应控制理论与应用. 哈尔滨工业大学出版社, 2005
11. 杨将新, 李华军, 刘冬骏. 单片机程序设计及应用. 电子工业出版社, 2006
12. 王书锋, 朱永红. 计算机控制技术. 华中科技大学出版社, 2011
13. 李均宜. 炉温仪表与热控制. 北京: 机械工业出版社, 1987
14. 常键生. 检测与转换技术. 北京: 机械工业出版社, 2001
15. 张锡富. 传感器. 第三版. 北京: 机械工业出版社, 2004
16. 胡寿松. 自动控制原理. 第五版. 北京: 科技出版社, 2004
17. 陶永华等. 新型 PID 控制及其应用. 北京: 机械工业出版社, 2000

