

陶瓷学院科技艺术学院

本科生毕业设计（论文）

中文题目： 烘箱温度计算机控制系统设计

英文题目： COMPUTER OVER TEMPERATURE CONTROL
SYSTEM DESIGN

院 系： 科技艺术学院

专 业： 自动化

姓 名： 马易俊

学 号： 201030455130

指导教师： 朱永红

完成时间： 2014-5-20

摘要

烘箱在我们的现实生活中有广泛的运用，例如在机械、化工、食品等各类工业中。它们都需要对烘箱的温度实现准确的控制。如果采用一般的控制方法难以实现对温度的准确控制。因此我们通过控制可控硅的通断从而控制加热装置开启或关闭，实现对烘箱温度的智能控制。

本设计采用 AT89C52 单片机为核心。介绍了烘箱温度控制系统的工作原理和软件设计的方法。该设计通过热电偶温度传感器对烘箱内的温度进行检测，再由 MAX6675 传感器对测得的信号进行放大和 A/D 转换，将模拟信号转换成数字信号输入到单片机中。通过显示电路将所测得的温度显示出来。如果温度超过了设定的温度值时，报警电路导通进行报警动作。如果温度低于所设定的温度值时，电热丝加热电路控制动作，对加热丝进行通电使得电热丝对烘箱进行加热。本设计实现了温度的采集、温度显示、温度控制等几个方面的内容。同时采用数码管的显示和键盘的输入实现人与单片机的交流，使得使用者对烘箱的温度进行直观的观察和方便的对温度进行控制。

关键词： AT89C52 单片机 温度控制 烘箱 MAX6675

ABSTRACT

Oven has widely used in our real life , such as machinery, chemicals, food and other industries. They are required for the oven temperature to achieve accurate control. If the control method is not generally used to achieve accurate control of the temperature . So we off by controlling the thyristor to control the heating device on or off, the oven temperature to achieve intelligent control .

This design uses AT89C52 micro-controller as the core . Describes the working principle of the method and software oven temperature control system design. The design is by thermocouple temperature sensor detects the temperature within the oven , and then measured by the MAX6675 sensor signal amplification and A / D converter converts the analog signal into a digital signal input to the MCU . The circuit shown by the measured temperature is displayed . If the temperature exceeds the temperature set value, the alarm circuit turns on the alarm action. If the temperature is below the set temperature value , the electric wire heating circuit control operation of the electric heating wire so that the wire is energized to heat the oven . The design and implementation of the temperature of the acquisition, temperature display, temperature control several aspects . While using the input digital tube display and keyboard man and micro-controller communication, allows the user to visually oven temperature observation and convenient temperature control.

KEYWORDS:AT89C52 micro-controller Temperature Control Oven
MAX6675

目录

摘要.....	I
ABSTRACT	II
1 引言.....	1
1.1 课题研究的意义.....	1
1.2 烘箱温度控制的发展状况.....	1
2 烘箱温度控制系统整体方案.....	3
3 烘箱系统硬件电路设计.....	4
3.1 AT89C52 芯片.....	4
3.2 温度控制电路设计.....	6
3.3 温度采集与信号转换电路设计.....	7
3.4 显示及键盘电路设计.....	9
3.5 时钟电路设计.....	错误!未定义书签。
3.6 复位电路设计.....	11
4 烘箱控制软件系统设计.....	12
4.1 软件设计总体框图设计.....	12
4.2 软件系统子程序设计.....	12
4.2.1 温度控制系统子程序.....	12
4.2.2 键盘输入系统软件设计.....	13
4.2.3 显示系统软件设计.....	14
5 结论.....	15
6 经济分析报告.....	16
6.1 目标成本.....	16
6.2 初步计算成本.....	16
6.3 经济分析及市场前景预算.....	16
致谢.....	18
参考文献.....	19
附件一 硬件总原理图.....	20
附件二 源程序.....	21

1 引言

1.1 课题研究的意义

在古代时期我们的祖先发现了火能够将食物烤熟，吃熟的食物能够让人不容易产生疾病。但那时候我们的祖先还不能理解为什么火能够将食物烤熟呢？通过前人的脚步我们了解到，这是温度的力量。温度是用来衡量物体的冷热程度的物理量，这只是一个抽象的概念，也是一个相对的概念。什么样的温度算高呢？什么样的温度算低呢？我们把冰水混合物的温度定以为零摄氏度。就拿这个作为衡量温度的标准，比这个温度高的我们将它定义为零上几摄氏度，比这个温度低的我们将它定义为零下几摄氏度。温度在我们的生活中无处不在，而且以不同的温度值存在。就拿水来说，有零摄氏度的水、有温水、有开水。对于我们人体而言也有很多不同的温度值，我们的外表皮的温度，血液的温度，心脏的温度都不相同。在我们当今的生活中对温度的控制也是一个至关重要的问题。例如加热食物，温度不能一直太高，如果温度太高就会将食物烧焦。也不能太低，如果温度过低食物将不会烧熟。加热的时间也有相关的要求，不能将食物加热太长时间，假设温度不是很高，但加热的时间足够长食物也将会烧焦。因此，在当今社会对温度的控制至关重要。

烘箱在我们的生活中运用非常的广泛。随着现代工业生产和生活的要求不断提高，对烘箱温度控制器的精度、通用性、安全稳定性等性能要求越来越高，同时其成本及实用性对人们来说也不容忽视。如果能将烘箱的温度控制的技术提高的话，那么人们的许多日常活动都将变得更加方便，对能源的节约会有非常重要的意义。

1.2 烘箱温度控制的发展状况

以前的烘箱采用的是手控式和电子式的温度控制装置，随着人们对温度控制精度和操作简便要求的提高，人们又发明了以单片机为核心的温度控制装置，该种装置是目前较先进的温度检测和控制装置。具有温度控制准确，操作简便和结构简单等众多的优点。因此现在大部分采用单片机对烘箱的温度进行控制，且越来越受到人们的喜爱。但同时它也还存在着许多的不足且不能满足人们某些方面的要求。例如其输出结果常用数码管显示，结果不够形象，对于复杂数据的处理

能力方面还存在着不足。

随着计算机技术的发展和完善，微型计算机也被广泛运用与检测和控制领域。微型计算机在测控技术中的运用，使得传统的测控手段，方法和设备发生了根本的变化，形成了自动化，实时化和智能化的微型计算机检测与控制系统。随着科学技术的发展和自动化程度的进一步提高，烘箱做为实验室的常用设备也在不断的发展，由原来单一的产品逐渐向多个方向发展，比如电脑控温、真空烘箱等等。

2 烘箱温度控制系统整体方案

在本设计课题中最主要考虑的问题是烘箱温度的控制。首先我们要选择利用什么控制装置对烘箱的温度进行控制。由于以单片机为核心的温度控制装置具有温度控制准确，操作简便和结构简单等众多优点，因此在这里采用以单片机为核心的控制装置对烘箱温度进行控制。对烘箱温度进行控制我们还需要考虑怎样将温度进行检测，怎样让单片机进行识别，怎样将温度信号进行显示以及怎样对温度信号进行控制等一些问题。为此需要设计以下几个模块，数据采集转换电路模块，控制电路模块，键盘显示模块，加热驱动模块，报警电路模块。对于各个模块需要完成的内容如下所示：

1. 数据采集模块主要是对数据采集，放大，转换模块完成温度信号的测量放大和转换。
2. 控制模块主要包括单片机系统，各种接口电路。
3. 显示模块主要是通过数码管显示来实现。
4. 加热驱动电路模块主要完成单片机控制可控硅加热。
5. 报警电路模块实现在烘箱温度超过设定温度时发出警告。

在本设计中采用 AT89C52 单片机作为控制系统的核心元件。根据以上的设计要求我们了解了基本的几个模块，根据模块的内容我们可以绘制出其原理图如图 2-1 所示：

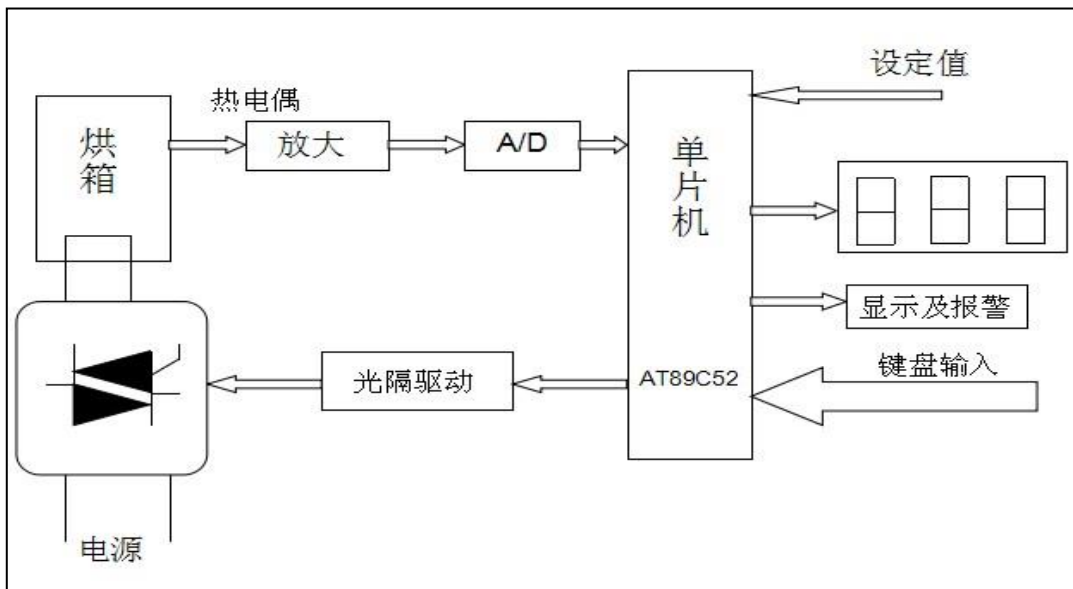


图 2-1 烘箱控制硬件原理图

3 烘箱系统硬件电路设计

3.1 AT89C52 芯片

1) AT89C52 芯片引脚如图 3—1 所示：

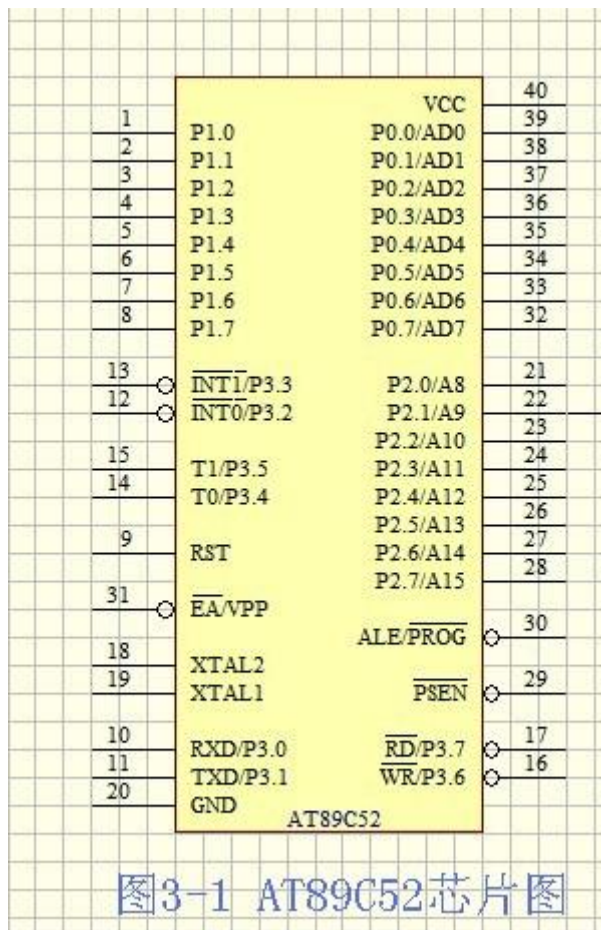


图3-1 AT89C52芯片图

2) AT89C52 单片机各引脚功能

(1) I/O 引脚 ($4 \times 8 = 32$): P0, P1, P2, P3。

1、P0 口：P0 口为一个 8 位漏级开路双向 I/O 口，每脚可吸收 8TTL 门流。当 P1 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/地址的第八位。在 FIASH 编程时，P0 口作为原码输入口，当 FIASH 进行校验时，P0 输出原码，此时 P0 外部必须被拉高。

2、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 口作为第八位地址接收。

3、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。

4、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 AT89C52 的一些特殊功能口，如下表 3—1 所示：

表 3—1 P3 口的第二功能

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

（2）控制引脚：

1、ALE——地址锁存使能。当访问外部存储器时，地址锁存允许的输出电平用于锁存地址的地位字节。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。

2、 $\overline{\text{PSEN}}$ ——外部程序存储器 ROM 的选通信号。在由外部程序存储器 ROM 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 RAM 时，这两次有效的 $\overline{\text{PSEN}}$ 信号将不出现。

3、 $\overline{\text{EA}}/\text{VPP}$ —— $\overline{\text{EA}}$ 访问外部程序存储器控制信号。当 $\overline{\text{EA}}$ 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， $\overline{\text{EA}}$ 将内部锁定为 RESET；当 $\overline{\text{EA}}$ 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）。

4、RSE——复位。当晶振在运行中，只要复位管脚出现 2 个机械周期高电平即可复位。内部有扩散电阻连接到 V_{ss} ，仅需要外接一个电容到 V_{cc} 即可实现上电复位。

（3）时钟晶体引脚：

1、XTAL1——反相振荡放大器输入和内部时钟发生电路输入。

2、XTAL2——反相振荡放大器输出。要使用外部时钟源驱动器件时，XTL2 可以不连接而由 XTL1 驱动。外部时钟信号无占空比的要求，因为时钟通过触发器二分频输入到内部时钟电路。

（4）电源引脚（2）

1、 V_{ss} ——地。

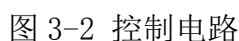
2、 V_{cc} ——电源。提供掉电，空闲，正常工作电压。

3.2 温度控制电路设计

控制部分最主要是电子开关电路。电子开关电路一般由光电耦合芯片、继电器、双向可控硅的等电子器件组成。电路要实现控制回路（输入）与负载回路（输出）之间的电隔离及信号耦合，可达到无触点，无火花接通和断开电器的目的。电子开关电路应用领域十分广泛，如用于计算机的接口，微机的测控系统，自动控制等领域。

本次毕业设计采用光电耦合器和双向可控硅等元器件构成开关电路，该电路除用于控制交流接触器，还可以用于控制灯光、加热器等，但一定要注意双向可控硅的参数是否能满足负载的要求。

工作原理：在输入端加一个控制信号，就可以控制输出端的“通”和“断”，



温度的测量方法有很多种，从测量是传感器中有无电信号可以划分为非电测量和电测量；从测量时传感器与被测对象的接触方式不同可以划分为接触式和非接触式。而每种测量方法所用到的测量仪器又有所不同，如金属热电偶温度计、热电阻温度计、热敏电阻温度计、光学温度计等。在本设计中我们采用热电偶进行测温，采用热电偶进行测温主要由于以下几个方面的内容。

1. 热电偶能将温度量转换成电量进行测量，所以对于温度的测量，控制和温度信号的放大，变换等都很方便。
2. 热电偶的结构简单，制造容易，价格便宜。
3. 惰性小，准确度高，测量范围广。
4. 能适应各种测温对象的要求，如点温和面温的测量。

第 7 页 共 34 页

连接部分的连接点分别为1点和2点。把1点的温度设为 $T_1^{\circ}\text{C}$ ，把2点的温度设为 $T_0^{\circ}\text{C}$ 。当两个接触点的温度不同时，对 $T_1^{\circ}\text{C}$ 温度端进行加热，当温度 $T_1 > T_0$ 时则在回路中就会产生热电动势，通过测量热电动势从而达到测温的目的。

对于不同的热电偶材料测量的温度范围和适用的环境会有所不同，这就需要我们z对热电偶制作材料有所要求。构成热电偶的两种材料A和B称为热电极，热电极主要由金属材料制成，有时也用非金属材料 and 半导体材料。对热电极主要有以下几个要求：必须要求材料的化学稳定性高，物理性能稳定，热电动势大及灵敏度高，线性度较好，价格便宜，机械性能好等。本设计采用铜—铜镍热电偶（T型）对温度进行测量，该种热电偶的正极为纯铜（TP），负极为铜镍合金（TN，康铜）。它的使用温度范围是 $-200^{\circ}\text{C} \sim 350^{\circ}\text{C}$ ，其在廉价金属热电偶中，它的准确度最高，。因此，这是一种最通用的热电偶，适用于一般的温度测量。在该函数中覆盖了 $-55^{\circ}\text{C} \sim +100^{\circ}\text{C}$ 的测温区，因此采用该种热电偶。

该设计的测温是采用T型热电偶进行，对温度信号进行采集后再通过MAX6675将模拟信号转化成数字信号输入到AT89C52单片机中。其与单片机的连接如图3-3所示：

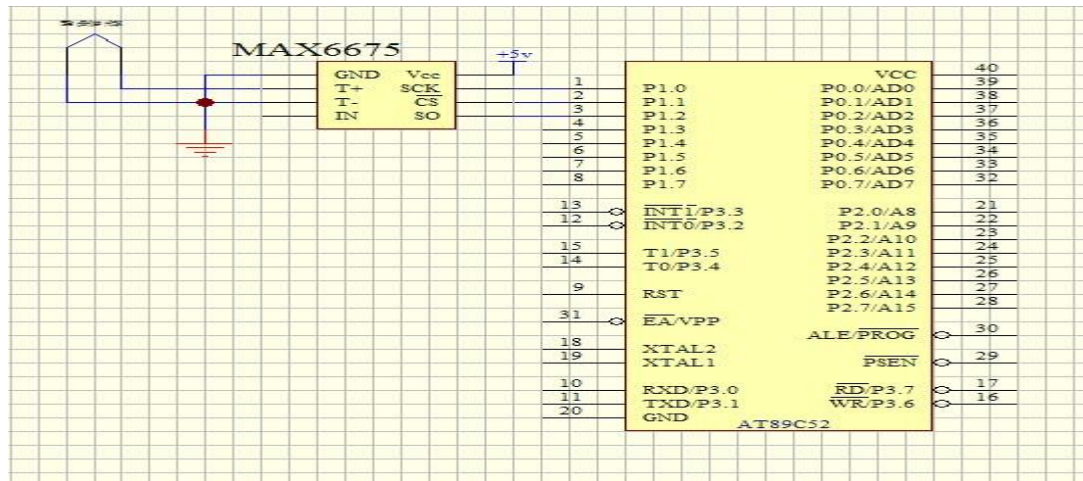


图3-3 测温电路

如图3-3所示，当单片机与CS相连的端口P1.1输出低电平且选用的P1.0口产生时钟脉冲连接SCK脚，使MAX6675的SO脚输出转换数据。在每一个脉冲信号的下降沿输出一个数据，16个脉冲信号完成一串完整的数据输出，先输出高电位D15，最后输出的是低电位D0，D14~D3为相应的温度转换数据。当P1.2为高电平MAX6675开始进行新的温度转换。

3.4 显示及键盘电路设计

本设计中的显示电路采用数码管显示，在数码管内部电路中一个数码管的引脚是 10 个，显示一个 8 字需要 7 个小段，另外还有一个小数点，所以在其内部一共有 8 个小的发光二极管，最后还有一个公共端。但是由于封装的方便单位数码管都封装成 10 个引脚。同时在数码管的连接方面也有不同，在它们的公共端我们将它分为共阳极和共阴极。对共阴极数码管而言其 8 个发光二极管的阴极连接在一起，且同时接地。而对共阳极数码管而言其 8 个发光二极管的阳极连接在一起，同时接 5V 的电源。一个数码管能显示 0~9 这 10 个数字，要是想显示小数点就导通 dp 这根数码管。数码管的内部结构和管脚如 3-4 图所示：

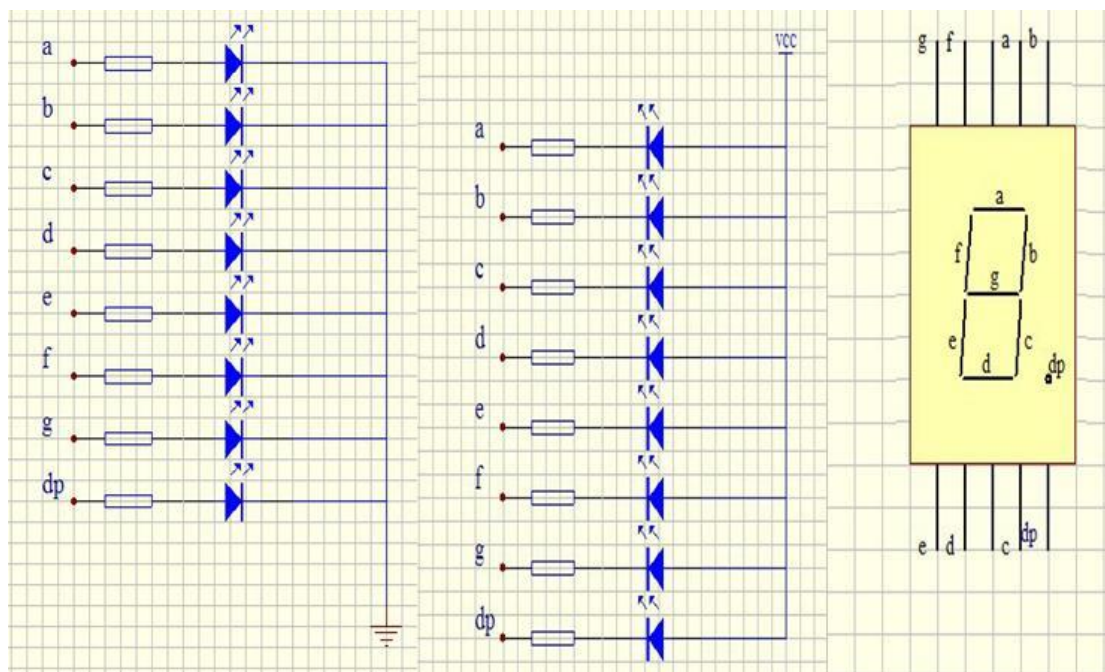


图 3-4 数码管内部结构和引脚

控制键盘设计采用多功能键的设计方法，节省了按键的数量，同时也简化了硬件线路，缩小了系统的规模。如下图所示，按键 B1，B2 既可对设定温度值进行加或减，也可对设定的时间进行加减。这样既可以减小系统的规模也可以使得系统的操作灵活简单。B3，B4 分别表示温度和时间的按钮。键盘电路如 3-4 图所示：

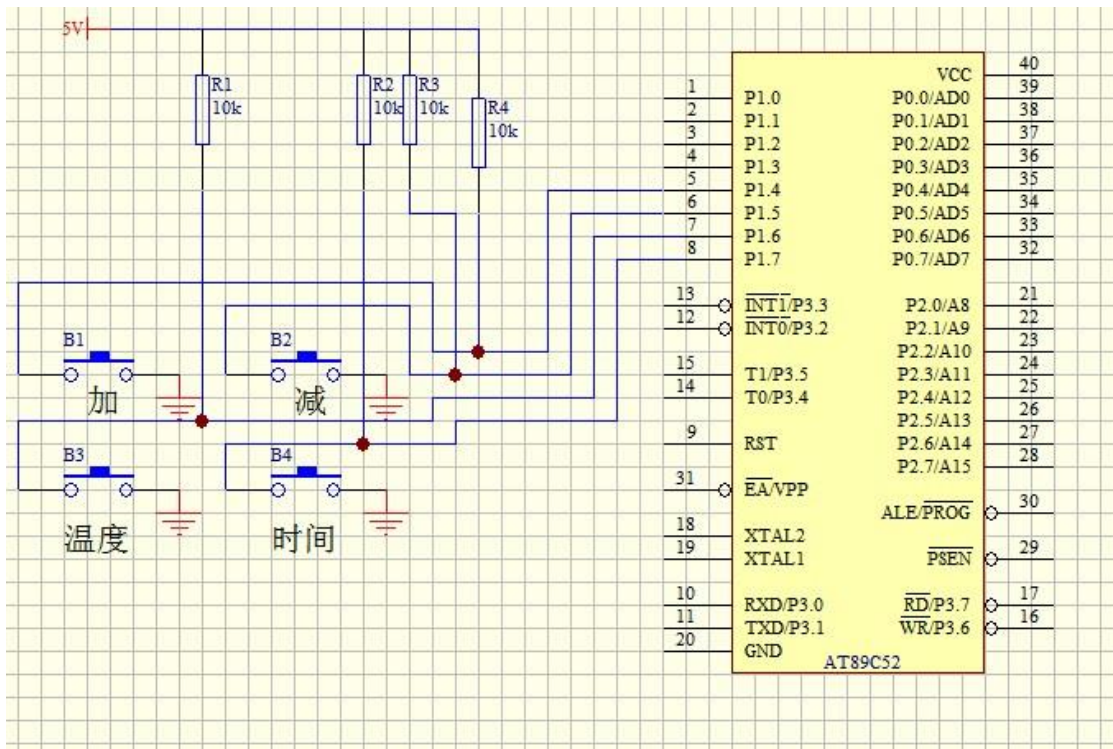


图 3-4 控制键盘

3.5 报警电路设计

本设计当温度超过设定温度时，这时就需要报警。电路中采用灯蜂鸣器进行报警。当 P2.1 为 0 时，报警灯亮；当 P2.1 为 1 时，报警灯不亮。如图 3-5 所示：

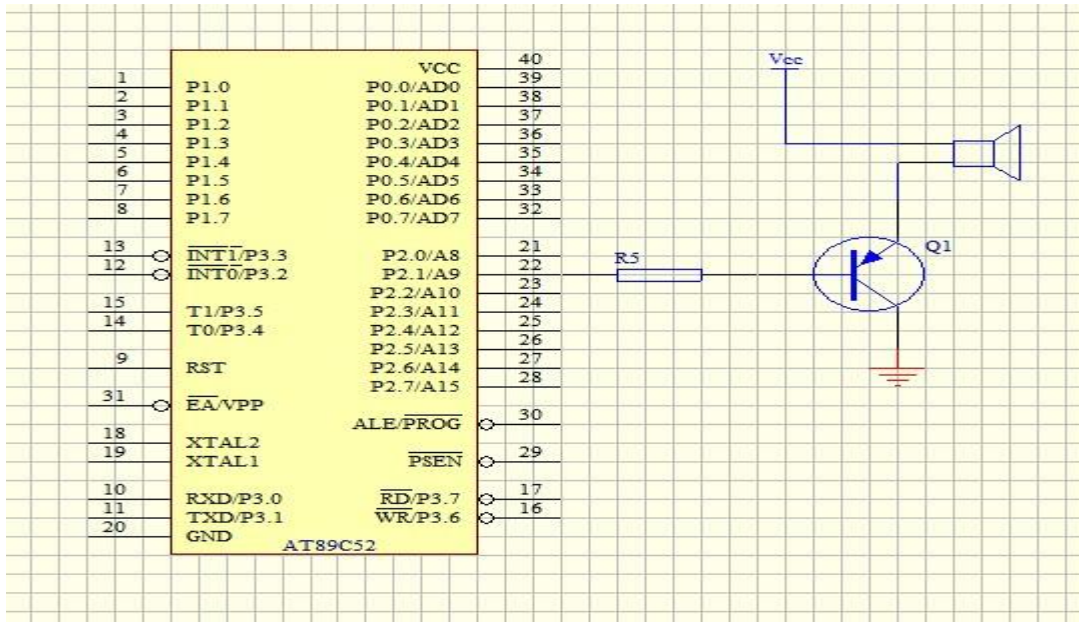


图 3-5 报警电路

3.6 复位电路设计

系统在启动运行时一般都需要复位,使中央处理器和系统中所用到的部件都处于一个确定的初始状态,并从这种状态开始工作。在该设计复位电路中将 RST 复位端接一个电容 C2 至 Vcc (电源)、在电容两端接一个按钮 S4、电阻 R4 一端与电容相连另一端至接 Vss(地),该电路就能执行复位操作。在上电的瞬间,电容通过电阻充电,就在端出现一定时间的高电平。只要保持 RST 引脚为高电平时间足够长,就可使 CPU 复位。所需高电平时间的长短与 Vcc 上升时间和振荡器起振时间有关。10MHz 时,约 1ms; 1MHz 时,约 10ms。若 Vcc 上升时间小于 20ms,那么从上电时间算起,只要保持 RST 引脚在高电平停留时间不小于 20ms 即可。该设计的复位电路如图 3-6 所示:

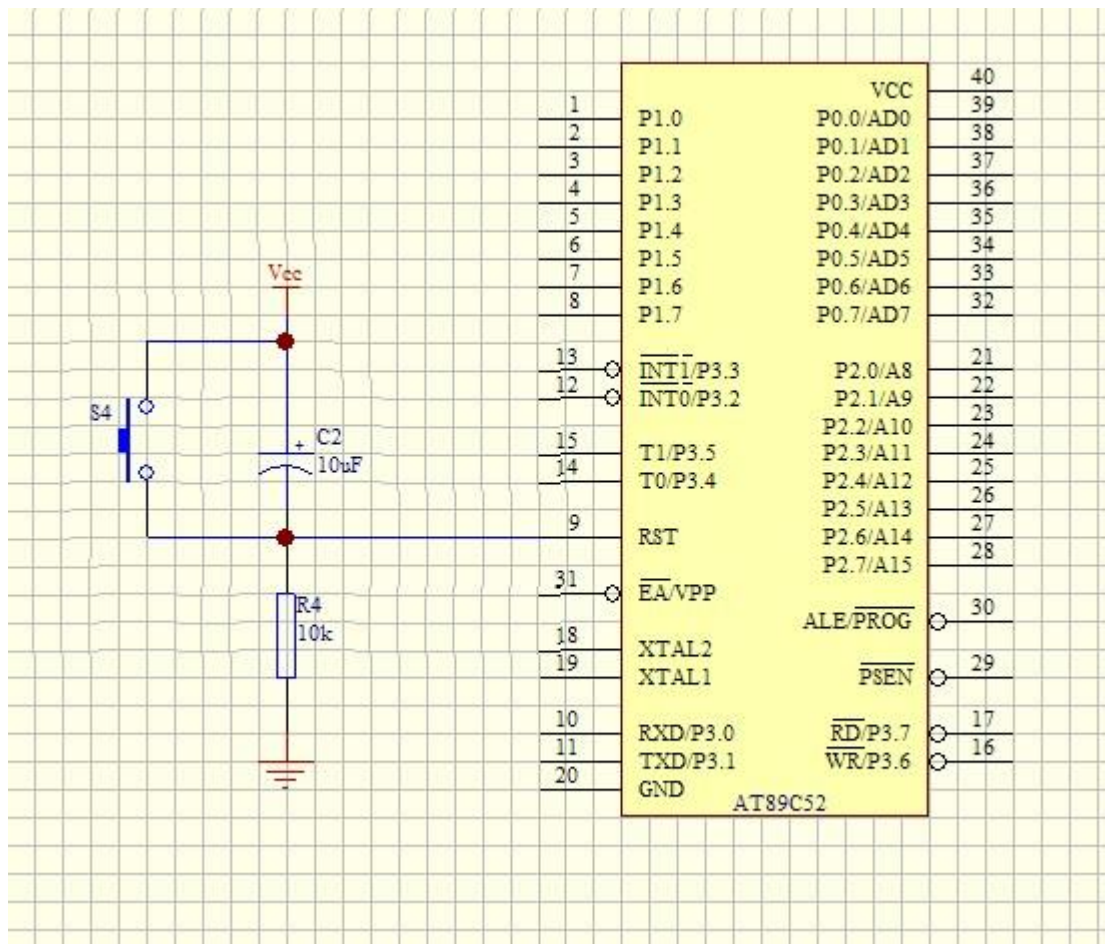


图 3-6 复位电路

根据以上的电路设计就可以组成一个整体的烘箱温度控制系统,该系统进行了温度的检测,信号的处理及显示,温度控制等操作。其总硬件原理图见附录一。

4 烘箱控制软件系统设计

4.1 软件设计总体框图设计

根据系统的总体设计要求以及硬件电路原理，按照硬件连接和各个模块芯片的特性以及功能实现要求，本设计的主要流程包括四个步骤：系统初始化，采样温度并送显，加热控制和报警。软件设计总程序框图如图4-1所示：

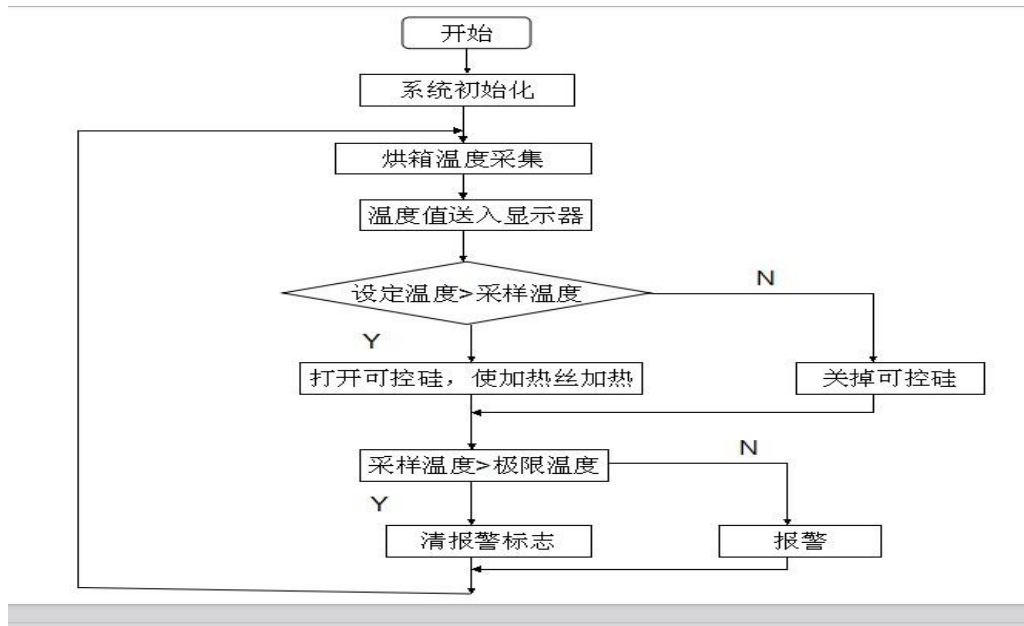


图 4-1 软件设计总程序图

4.2 软件系统子程序设计

4.2.1 温度控制系统子程序

对烘箱温度进行控制时首先将模块进行初始化，由 A/D 转换后显示烘箱的初始温度，再通过按键设置所要控制温度，当实际温度值大于所设定的温度值时热电丝就停止工作，当实际温度值小于所设定的温度值时热电丝就开始工作。其基本流程如图 4-2 所示：

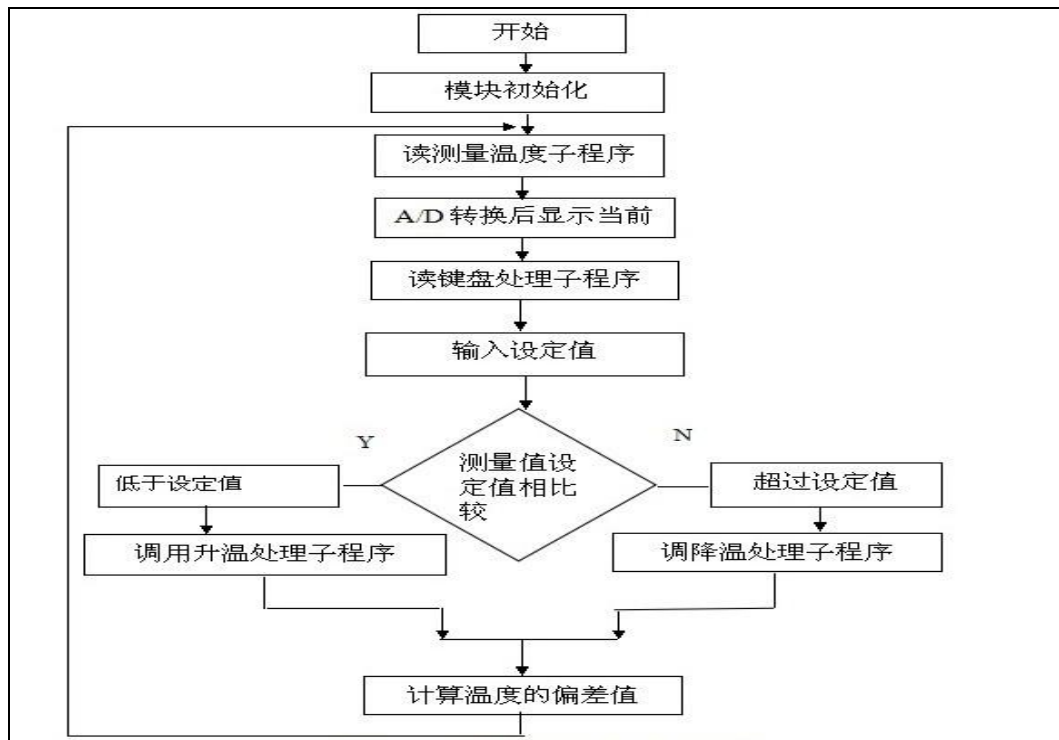


图 4-2 温度控制系统流程图

4.2.2 键盘输入系统软件设计

本设计的键盘是独立式键盘，采用键盘扫描法进行按键扫描。按键选择子程序采用查询方式来实现按键的识别，在查询过程中 CPU 只要一有空闲就会调用键盘扫描程序进行扫描。从而对键盘进行查询和识别，并予以处理。按键扫描流程如图 5-2 所示：

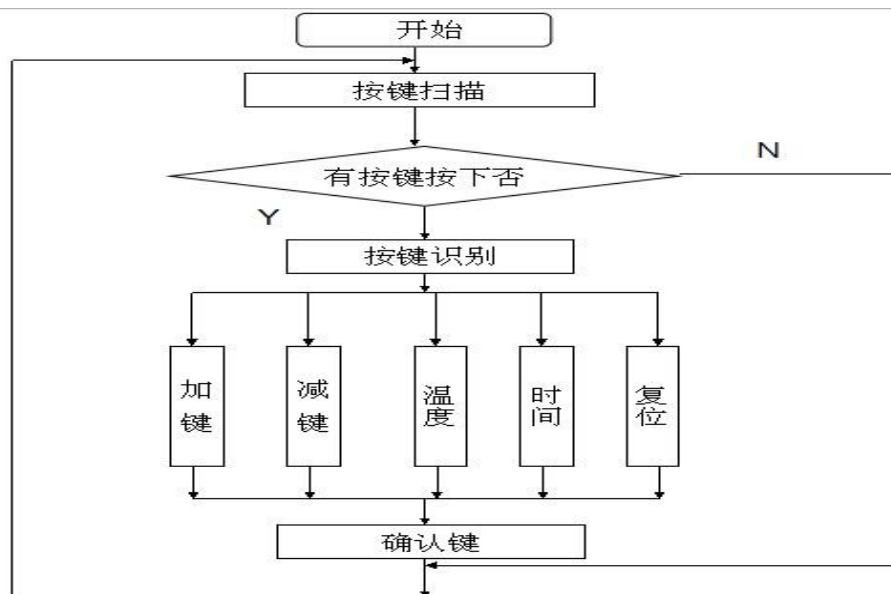


图4-3 按键扫描流程图

4.2.3 显示系统软件设计

本设计用了 4 个七段共阳极数码管，首先进行开始操作取用要显示的数送位段码到输出，通过延时将四个数据显示完，再进行循环。其显示电路流程图如图 4-4 所示：

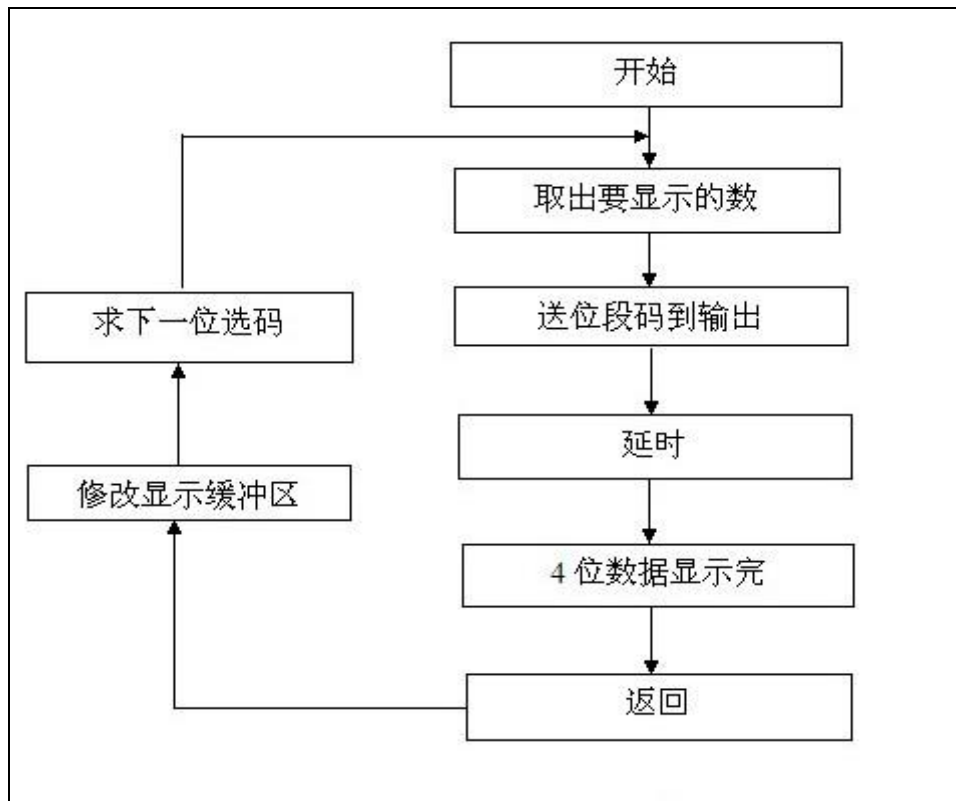


图 4-4 显示电路流程图

5 结论

由于受到资金及技术水平的限制，我国烘箱温度控制系统自动化程度较低，这种状况既严重制约了很多设备精度水平的提高，又造成了较大的能源浪费。因此，为了改变这一现状，本设计针对烘箱温度控制问题进行了研究。

本设计围绕烘箱温度控制系统设计展开，在深入了解烘箱温度控制系统的基础上，分别对烘箱温度控制系统的整体架构、控制部分进行了设计与实现。该烘箱温度控制系统能够满足设计要求，运行稳定、功能先进、操作方便，对于今后类似系统的设计、开发具有一定的指导意义。本设计采用 AT89C52 单片机为核心。介绍了烘箱温度控制系统的工作原理和软件设计的方法。该设计通过热电偶温度传感器对烘箱内的温度进行检测，再由 MAX6675 传感器对测得的信号进行放大和 A/D 转换，将模拟信号转换成数字信号输入到单片机中。通过显示电路将所测得的温度显示出来。如果温度超过了设定的温度值时，报警电路导通进行报警动作。如果温度低于所设定的温度值时，电热丝加热电路控制动作，对加热丝进行通电使得电热丝对烘箱进行加热。

本设计虽然在烘箱温度控制系统的设计开展了一些有意义的研究工作，但是由于时间和条件的限制，本设计的研究工作还不够深入和系统。希望在以后的研究过程中能将其功能更加完善。

6 经济分析报告

每一种新产品的设计和开发，大部分是基于该产品能带来更大的收益。要获得更大的收益，就需要得到购买者的认可，这就是性价比的问题了，也就是说如何用最少的钱买到最好的产品。那么对于生产厂家而言，怎样使用最小的成本生产出最有市场价值的产品无疑是设计过程中非常重要的问题。所以，在设计中，如何通过设计来减少材料的使用和提高材料的利用率，并通过整体设计使产品的性能价格比最优化，是设计者必须重视的问题。

本设计对温度控制系统进行了设计研究。

6.1 目标成本

单位产品目标成本=预测价格×（1-税率-目标利润率）

此处税率=8%， 目标利润率=50%

6.2 初步计算成本

根据本设计的系统所需的材料费用、工资、折旧等消耗，定价参考目前市场价格，得出了成本概算见表 6—1：

表 6—1 温度控制系统成本概算表（单位：元）：

序号	项目	金额	备注
1	微处理器	25.00	AT89C52 单片机
2	显示器	20.00	7 段数码管
4	温度传感器	16.80	铂铑—铂铑热电偶传感器
5	集成芯片	10.00	MAX6675
6	光电耦合器	25.00	MOC3021
7	报警器	5.00	蜂鸣器
8	其余材料	20.00	其余零散材料
总计	121.8		产品销售费用/个

6.3 经济分析及市场前景预算

一个好的应用方案，不仅要看技术含量，还要看它是否实用，性价比是不是

适当。本次设计的温度控制系统无论从技术含量还是实际性上来看都比较好。由于本次设计的温度控制系统操作方便，安全，对硬件以及技术要求不高，易于维修，所以便于多数中小型企业的采用，使资金回炉快，便于大规模投产。所以此设计是可行的。

致谢

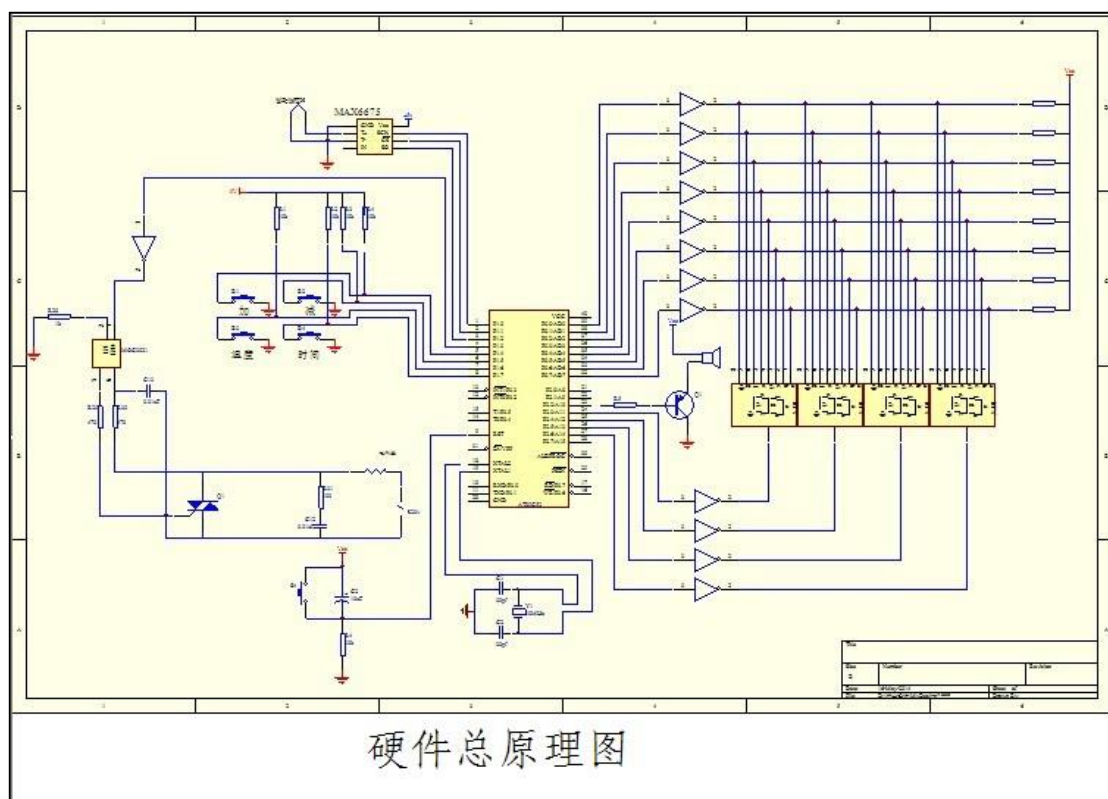
在这里我首先要感谢的是我的指导老师朱永红老师，他在我们的毕设过程中给了我们很大的指导和帮助。在我们遇到困难的时候总是耐心的帮助我们分析错误，理清我们的思路，并且教给我们很多调试程序的经验和实际设计过程中应注意的问题，对我的设计提出了许多宝贵的意见，使我们能在规定时间内保质保量地完成了毕业设计。在此对他给予的帮助表示衷心的感谢！

同时我还要感谢给我提供帮助的老师和同学们，通过他们的帮助我解决了很多在该设计中遇到的很多问题。是他们给了我一些宝贵的建议，让我了解了大概的设计思路。在使用一些软件的时候有一些不懂的问题是帮助他们解决问题。对此我想说一句：“谢谢！我敬爱的老师和亲爱的同学们”。正因为老师和同学的帮助我才能顺利的完成这次设计。

参考文献

- [1] 盛珣华, 李润梅编著. 计算机控制系统[M]. 北京:清华大学出版社; 北京交通大学出版社, 2007: 252-258.
- [2] 潘新明, 王燕芳编著. 微型计算机控制技术[M]. 北京: 电子工业出版社, 2003: 23-61.
- [3] 姜忠良, 陈秀云编著. 温度的测量与控制[M]. 北京: 清华大学出版社, 2005: 60-95.
- [4] 王魁汉等编著. 温度测量实用技术[M]. 北京: 机械工业出版社, 2006:84-126.
- [5] 章彬宏, 吴青萍主编. 模拟电子技术[M]. 北京: 北京理工大学出版社, 2008:32-62.
- [6] 王义军主编. 数字电子技术基础[M]. 北京: 中国电力出版社, 2007:125-134
- [8] 杨鹏主编. 计算机控制系统. 北京: 机械工业出版社, 2008
- [9] 沈红卫主编. 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
- [10] 余永权等主编. 单片机在控制系统中的应用. 北京: 电子工业出版社, 2003
- [11] 汉才主编. 单片机原理及其接口技术. 北京: 清华大学出版社, 1996
- [12] 王兆安, 黄俊主编. 电力电子技术. 北京: 机械工业出版社, 2000
- [13] 于海生主编. 微型计算机控制技术. 北京: 清华大学出版社, 1999
- [15] 张俊谟主编. 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, 20001

附件一 硬件总原理图



附件二 源程序

```
adc_read    data 30h      ;采样数据暂存单元
adc_BCD     data 31h      ; 采样值转化成温度值存储单元
disp_flash  data 32h      ; 闪烁控制位单元
T2CON       data 0c8H
RCAP2L      data 0caH
RCAP2H      data 0cbH
T11         data 0ccH
TH2         data 0cdH
flash_bit   data 33h
key_touch   bit  00H      ; 按键标志位
flash       bit  01h
flash1      bit  02h
led_alarm   bit p3.4
out_si      bit p3.5

ORG  0000H
LJMP  MAIN
ORG  000BH
LJMP  INTT0
ORG  0100H

MAIN:      mov sp, #67h
           mov r0, #7fh
lp1:       mov @r0, #00h      ; 工作单元清 0
           djnz r0, lp1
           mov @r0, #00h
           mov T2CON, #30H    ; 串口初始化, 并规定波特率
```

```
mov TH2, #0FFH
MOV TL2, #0ECH
MOV RCAP2H, #0FFH
MOV RCAP2L, #0ECH
MOV SCON, #50H
MOV PCON, #00H
ORL T2CON, #04H
MOV TMOD, #00000001B      ; 定时器 0 初始化
MOV TH0, #00H
MOV TL0, #00H
MOV IE, #10000010B        ; 开中断
clr TR0
MOV 43H, #10
MOV 44H, #10
MOV 45H, #0
MOV 46H, #0
MOV 47H, #0
WAIT: mov r2, #2fh
lpp:  lcall DEALYY
      LCALL KEY_DEAL
      DJNZ R2, lpp
      lcall adc0804
      mov a, p0
      mov adc_read, a
      LCALL ADC_TEMP
      mov a, adc_BCD
      mov b, #100
      div ab
      mov a, b
```

```
        mov b, #10
        div ab
        mov 41h, a
        mov 42h, b
        lcall disp
        jb tr0, lp2
        lcall t_control
        lcall alarm
lp2:      mov a, adc_bcd
        MOV SBUF, A
        jnb ti, $
        clr  ti
        ljmp WAIT

t_control: mov r1, 45h
        cjne r1, #00, t4
        mov a, #00h
        ljmp t5
t4:      mov a, #00h
        mov b, #100
t3:      add a, b
        djnz r1, t3
t5:      push acc
        mov a, 46h
        mov b, #10
        mul ab
        mov r1, a
        pop acc
```

```
        add a, r1
        add a, 47h
        cjne a, adc_bcd, $+1
        jc t1
        clr out_si
        ret
t1:      setb out_si
        ret

alarm:   mov a, adc_bcd
        cjne a, #80, $+1
        jc t2
        clr led_alarm
        ret
t2:      setb led_alarm
        ret

INTT0:   CLR EA
        PUSH ACC
        PUSH PSW
        SETB RS0
        CLR RS1
        INC DISP_flash
        MOV A, DISP_flash
        CJNE A, #03H, $+1
        JC INTO_RET
        MOV DISP_flash, #00H
        cpl flash
INTO_RET: MOV TH0, #00H
```

MOV TL0, #00H

POP PSW

POP ACC

SETB EA

RETI

ADC_TEMP: MOV A, ADC_READ

push acc

mov b, #05

div ab

mov b, a

pop acc

add a, b

mov adc_bcd, a

ret

adc0804: mov dptr, #0feffh

clr p2.0

MOVX @DPTR, a

setb P2.0

LCALL DEALY

clr p2.0

MOVX a, @DPTR

mov adc_read, a

setb p2.0

ret

DEALY: MOV R7, #0AH

DEALY1: MOV R6, #0FFH

```
DEALY2:    DJNZ R6, DEALY2
           DJNZ R7, DEALY1
           RET
```

```
DISP:      jnb flash, l1
           setb flash1
           mov a, flash_bit
           add a, #40h
           push acc
           mov r1, a
           mov a, @r1
           push acc
           mov @r1, #10
```

```
L1:        MOV DPTR, #PTMB
           MOV R0, #40H
           MOV R6, #08H
```

```
14:        MOV A, @R0
           MOVC A, @A+DPTR
           MOV R7, #08H
```

```
L2:        CLR P1.6
           NOP
           RLC A
           MOV P1.7, C
           SETB P1.6
           DJNZ R7, L2
           INC R0
           DJNZ R6, L4
           jnb flash1, l3
           pop acc
```

```
        mov r2, a
        pop acc
        mov r1, a
        mov a, r2
        mov @r1, a
        clr flash1
13:      RET

DEALYY:  MOV R7, #0AH
DEALY11: MOV R6, #0FFH
DEALY22: DJNZ R6, DEALY22
        DJNZ R7, DEALY11
        RET

KEY_DEAL: MOV A, P1
        CPL A
        ANL A, #3FH
        JNZ KEY1
KEY0:    CLR KEY_TOUCH
KEY2:    RET
KEY1:    JB KEY_TOUCH, KEY2
        LCALL DEALYY
        MOV A, P1
        SETB KEY_TOUCH
        JNB ACC. 0, SK1
        JNB ACC. 1, SK2
        JNB ACC. 2, SK3
        JNB ACC. 3, SK4
        JNB ACC. 4, SK5
```



```
JNB ACC. 5, SK6
LJMP KEY0
SK1:    setb tr0
        mov flash_bit, #07h
        RET
SK2:    dec flash_bit
        mov a, flash_bit
        cjne a, #05h, $+1
        jc pp4
        ljmp pp5
pp4:    mov flash_bit, #07h
pp5:    RET
SK3:    mov a, flash_bit
        add a, #40h
        mov r0, a
        inc @r0
        mov a, @r0
        cjne a, #0ah, pp2
        mov @r0, #00h
pp2:    RET
SK4:    mov a, flash_bit
        add a, #40h
        mov r0, a
        dec @r0
        mov a, @r0
        cjne a, #0ffh, pp3
        mov @r0, #09h
pp3:    RET
SK5:    clr tr0
```

```
        clr flash
        mov flash_bit, #0ffh
        RET
SK6:    RET

PTMB:DB 0C0H, 0CFH, 0A4H, 0B0H, 99H, 92H
        DB 82H, 0F8H, 80H, 90H, 0ffh
end
```