摘要

温度是工业控制中主要的被控参数之一。随着电子技术和计算机技术的迅速发展，微机测量和控制技术得到了迅速的发展和广泛的应用。单片机具有处理能力强、运行速度快、功耗低等优点，尤其在温度测量和控制方面，控制简单方便，测量范围广，精度较高，得到广泛应用。

本设计以STC89C52RC单片机为核心的温度控制系统的工作原理和设计方法。温度信号由温度芯片DS18B20采集，并以数字信号的方式传送给单片机。文中介绍了该控制系统的硬件部分，包括：温度检测电路、温度控制电路、显示电路。单片机通过对信号进行相应处理，从而实现温度控制的目的。文中还着重介绍了软件设计部分，在这里采用模块化结构，主要模块有：LCD1602显示程序、键盘扫描及按键处理程序、温度信号处理程序、继电器控制程序、超温报警程序。

关键词：STC89C52RC，温度，报警，DS18B20

1 绪 论

温度控制系统广泛应用于社会生活的各个领域 ,如家电、汽车、材料、电力电子等 ,常用的控制电路根据应用场合和所要求的性能指标有所不同 , 在工业企业中,如何提高温度控制对象的运行性能一直以来都是控制人员和现场技术人员努力解决的问题。这类控制对象惯性大,滞后现象严重,存在很多不确定的因素,难以建立精确的数学模型,从而导致控制系统性能不佳,甚至出现控制不稳定、失控现象。传统的继电器调温电路简单实用 ,但由于继电器动作频繁 ,可能会因触点不良而影响正常工作。控制领域还大量采用传统的PID控制方式,但PID控制对象的模型难以建立,并且当扰动因素不明确时,参数调整不便仍是普遍存在的问题。而采用数字温度传感器DS18B20，因其内部集成了A/D转换器，使得电路结构更加简单，而且减少了温度测量转换时的精度损失，使得测量温度更加精确。数字温传感器DS18B20只用一个引脚即可与单片机进行通信，大大减少了接线的麻烦，使得单片机更加具有扩展性。由于DS18B20芯片的小型化，更加可以通过单跳数据线就可以和主电路连接，故可以把数字温度传感器DS18B20做成探头，探入到狭小的地方，增加了实用性。更能串接多个数字温度传感器DS18B20进行范围的温度检测。

1.1温度控制系统的简要介绍

温度控制系统是以温度为主要的控制变量。系统主要是通过温度传感器对工业现场的温度变化引起的其他物理量变化进行测量，然后通过电路转化为数字量转送到单片机中。在单片机对温度数据进行处理之后，根据用户的设定对加热电路或制冷电路进行控制，从而达到控制调节温度的目的。控制系统主要由控制器和控制对象两部分组成，通过一定的控制方法使系统达到所要求的控制性能。控制模式有开环控制、闭环控制和复合控制三种。所谓的开环控制是控制器与控制对象之间只有正向作用，没有反相联系，是一种单向的控制过程。如果控制器与控制对象之间既有正向作用又有反相联系，这种控制方式称为闭环控制或反馈控制。在某种情况下，为了达到较好的控制效果，往往将开环控制和闭环控制结合起来，这种控制方式称为复合控制。

大部分温度控制系统采用典型的负反馈式温度控制系统，系统基本框图1-1所示。其中数字控制器的功能由微型计算机实现。

1.2温度控制系统在国内外的发展概况

1.2.1 国外温度控制系统

自70年代以来，由于工业过程控制的需要，特别是在微电子技术和计算机技术的迅猛发展及自动控制理论和设计方法发展的推动下，国外温度控制系统发展迅速，并在智能化、自适应、参数自整定等方面取得成果，在这方面，以日本、美国、德国、瑞典等国技术领先，都生产出了一批商品化的、性能优异的温度控制器及仪器仪表，并在各行业广泛应用。它们主要具有如下特点：

1）适应于大惯性、大滞后性等复杂温度控制系统的控制。

2）能够适应于受控系统数学模型难以建立的温度控制系统的控制。

3）能够适应于受控系统过程复杂、参数时变的温度控制系统的控制。

4）这些温度控制系统普遍采用自适应控制、自校正控制、，模糊控制、人工智能等理论及计算机技术，运用先进的算法，适用的范围广泛。

5）普遍温控器具有参数自整定功能。借助计算机软件技术，温控器具有对控制对象控制参数及特性进行自动整定的功能。有的还具有自学习功能，它能够根据历史经验及控制对象的变化情况，自动调整相关控制参数，以保证控制效果的最优化。

6）温度控制系统具有控制精度高、抗干扰能力强的特点。

目前，国外温度控制系统及仪表正朝着高精度、智能化、小型化等方面快速发展。

1.2.2 国内温度控制系统

我国对模糊控制理论的研究与应用起步比较晚，然而发展很快，在各个领域取得了许多有影响的成果。诸如在模糊控制、模糊辨识、模糊聚类分析、模糊图像处理、模糊集合论、模糊模式识别等领域取得了不少有实际影响的结果。1988年哈尔滨工业大学在酒精厂10T/H工业染料煤链条路上实现鼓风量-蒸汽压力Fuzzy-PI双模控制。1990年模糊控制应用于电场过热汽温回路的自动控制。1994年模糊控制成功应用于镇海发电厂200MW机组的主蒸汽压力及温度自控系统。

目前，我国在温度等控制仪表业与国外的差距主要表现在如下几个方面：

1）行业内企业规模小，且较为分散，造成技术力量不集中，导致研发能力不强，制约技术发展。

2）商品化产品以PID控制器为主，智能化仪表少，这方面同外国差距较大。目前，国内企业复杂的及精度要求高的温度控制系统大多采用进口温度控制仪表。

3）仪表控制用关键技术、相关算法及控制软件方面的研究较国外滞后。例如：在仪表控制参数的自整定方面，国外已有较多的成熟产品，但由于国外技术保密及我国开发工作的滞后，还没有开发出性能可靠的自整定软件。控制参数大多数靠人工经验及现场调试来确定。这些差距，是我们必须努力克服的。

随着我国经济的发展及加入WTO，我国政府及企业对此都非常重视，对相关企业资源进行了重组，相继建立了一些国家、企业的研发中心，并通过合资、技术合作等方式，组建了一批合资、合作及独资企业，使我国温度等仪表工业得到迅速的发展。

2设计方案的选择与论证

温度控制有许多种方法，可供选择的器件和运用的技术也有多种，因此，系统的总体方案设计应在满足系统整体性能指标的前提下，充分考虑系统使用的环境，所选的结构要尽量简单实用，易于实现，器件的选用要着眼于合适的参数、稳定的性能、较低的功耗、低廉的成本以及较好的性能等等。本章将介绍基于单片机的测控系统的总体方案的设计并给出结构框图，分别从软硬件两个方面来进行讨论。

2.1 系统总体方案设计

温控系统是将负载的电加热丝看成是被控对象，温度是控制量,控制前先设定温度值，然后对电加热丝进行温度采集，并将采集的数据通过串行通信传送给上位机，通过控制P口的高低电平，控制电加热丝的通断，从而实现对温度的控制。

此课题是一个基于单片机的温度测控系统，主要由单片机构成温度采集与控制终端，实现温度采集与控制。此系统由软硬件两部分组成，硬件电路是系统的结构框架，是软件的载体，软件是系统的内核，通过硬件来进行具体操作，因此软硬件相互配合，共同完成各种功能。此系统硬件主要由以下几个部分构成：单片机、温度采集电路、温度控制电路、LCD显示电路，而系统软件主要通过对单片机编程来实现，此系统使用C语言来完成编程。

2.2硬件总体方案设计

本文所研究的系统硬件部分按照功能大致分为以下几个部分：最小系统、测温部分、控温部分以及LCD1602显示部分。其结构框图如图2.1所示：

单片机

温度采集

温度控制

按键电路

LCD1602

受

控

对

象

本设计本着低功耗、低成本、性能好等原则，最终选用以下器件来搭建硬件平台：

1.单片机最小系统

单片机最小系统包括单片机芯片，时钟电路和复位电路。时钟电路用于产生单片机工作时所必须的时钟信号。STC89C52单片机的内部电路在时钟信号控制下，严格地按时序执行指令进行工作；复位操作是单片机的初始化操作，只需给单片机的复位引脚RST加上大于2个机器周期的高电平就可以使单片机复位。

2.测温部分

传统的温度检测大多以热敏电阻为温度传感器，这种热敏传感器的工作原理是其电阻值随着温度的变化而发生显著变化，热敏传感器广泛用于一般精度的温度测量，或在计量设备、晶体管电路中作温度补偿。由于热敏电阻的可靠性差，测量温度准确率低，而且必须经过专门的接口电路转换为数字信号后才能由单片机进行处理，在高精度要求的温度检测应用中，热敏电阻已经被精度高、准确性好的各种集成温度采集设备所代替。

本部分采用的是DS18B20传感器进行温度测量，它在测量温度、转换时间、传输距离、分辨率等方面也有很大的改进。与传统的热敏电阻相比，它不仅能够直接读出被测温度值，并且可以根据实际要求通过简单的变成来读取9到12位的温度值，DS18B20温度变换功率来源于数据总线，总线本身也可以向所挂接的DS18B20供电，而无需额外电源，因而使用DS18B20可使系统结构更简单，可靠性更高，因此，DS18B20被广泛应用于温度采集与处理、数字温度计及各种温度控制系统中。

3.控制电路部分

该部分通过单片机的P口输出的高低电平来控制固态继电器的通断，从而决定电热丝是否加热，当P口输出低电平时，加热电阻通电，周围的温度缓慢升高，DS18B20测得的温度值也升高；当P口输出高电平时，加热电路断开，温度回落。

4报警电路

通过单片机io口控制蜂鸣器，来报警但温度低于或是高于设置温度时候，单片机的io口拉低，这时候驱动继电器的三极管就会工作，起到报警的作用

2.3 软件总体方案设计

温度控制系统的硬件电路确定之后，其主要功能将依赖于软件来实现，本系统的软件主要是完成温度数据的采集并把采集的数据通过串口通信传送给上位机，同时接收上位机的命令，达到温度控制的目的。软件具体结构框图如图2.2所示：

开始

初始化

采集温度

判断温度与设定值

继电器控制

执行相应的模块

3 个各单元电路的设计

3.1 主控电路

一般在系统的设计当中，能否完成设计任务最重要的就在于系统的核心器件是否选择合适，而单片机更是是系统控制的核心，所以对单片机的选择更是异常重要。如果选择了一个合适的单片机不仅可以最大地简化系统的操作，而且其功能可能是最好的，可靠性也比较高，对整个系统来说更方便。目前，市面上的单片机的种类繁多，并且他们在功能方面也是各自有各自的特点。在一般的情况下来讲，在选择单片机时要需要考虑的几个方面有：

（1）单片机最基本性能参数指标。例如：执行一条指令的速度、程序存储器的容量，I/O口的引脚数量等。

（2）单片机的某些增强的功能。

（3）单片机的存储介质。例如：对于程序存储器来说，最好选用的是Flash的存储器。

（4）单片机的封装形式。封装的形式多种多样，例如：双列直插封装、PLCC封装及表面贴附等。

（5）单片机对工作的温度范围的要求。例如：在进行设计户外的产品时，就必须要选用工业级的芯片，以达到温度范围的要求。

（6）单片机的功耗。例如，如果信号线取电只能提供几mA的电流，所以为了能满足低功耗的要求这个时候选用STC的单片机是最合适的。

（7）单片机在市面上的销售渠道是否畅通、其价格是否便宜。

（8）单片机技术的支持网站如何，卖家提供的芯片资料是否足够完善，是否包含了用户手册，设计方案举例，相关范例程序等。

（9）单片机的保密性是否很好，单片机的抗干扰的性能如何等。

51系列单片机它在指令系统、硬件结构和片内资源等方面与标准的52系列的单片机可以完全的兼容。51系列的单片机执行速率快(最高时钟频率为90 MHz)，功耗低，在系统、在应用可编程,不占用用户的资源。根据本系统设计的实际要求，选择STC89C52RC单片机做为本设计的单片机使用，它是由ATMEL公司生产的高性能、低功耗的CMOS 8位单片机。STC89C52RC单片机具有以下几个性能特点：4 k字节的闪存片内程序存储器，128字节的数据存储器，32个外部输入和输出口，2个全双工串行通信口，看门狗电路，5个中断源，2个16位可编程定时计数器，片内震荡和时钟电路且全静态工作并由低功耗的闲置和掉电模式。

单片机的引脚功能图如图3.1所示。

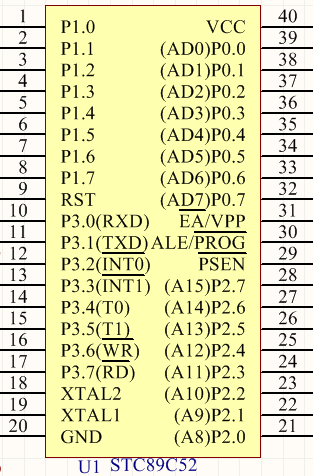


图3-1 51单片机的引脚功能图

3.1.1单片机引脚功能

（1）电源引脚

Vcc（40脚）：正电源的引脚，工作电压是5 V。

GND（20脚）：接地端。

（2）时钟电路的引脚XTAL1和XTAL2

为了产生时钟信号，在STC89C52RC单片机的芯片内部已经设置了一个反相放大器，其中XTAL1端口就是片内反相放大器的输入端，XTAL2端则是片内振荡器反相放大器的输出端 [5]。单片机使用的工作方式是自激振荡的方式，XTAL1和XTAL2外接的是12 MHz的石英晶振，使内部振荡器按照石英晶振的频率频率进行振荡，从而就可以产生时钟信号。时钟信号电路如图 4.2所示。

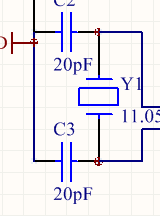


图4-2 时钟信号电路

（3）复位RST（9脚）

当振荡器运行时，只要有有两个机器周期即24个振荡周期以上的高电平在这个引脚出现时，那么就将会使单片机复位，如果将这个引脚保持高电平，那么51单片机芯片就会循环不断地进行复位。复位后的P0口至P3口均置于高电平，这时程序计数器和特殊功能寄存器将全部清零。本课题设计的单片机复位电路如图4.3所示。

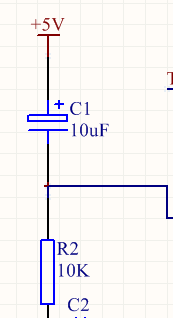


图4-3 单片机复位电路图

（4）输入输出口（I/O口）引脚

P0口是一个三态的双向口，既可以作为数据和地址的分时复用口，又可以作为通用输入输出口[5]。P0口在有外部扩展存储器时将会被作为地址/数据总线口，此时P0口就是一个真正的双向口；而在没有外部扩展存储器时，P0口也可以作为通用的I/O接口使用，但此时只是一个准双向口；另外，P0口的输出级具有驱动8个LSTTL负载的能力即输出电流不小于800 uA。

P1口是一个带内部上拉电阻的8位双向I/O口，而P1口只有通用I/O接口一种功能，而且P1口能驱动4个LSTTL负载；在使用时通常不需要外接上拉电阻就能够直接驱动发光二极管；在端口置1时，其内部上拉电阻将端口拉到高电平，作输入端口用。

对于输出功能，在单片机工作的时候，可以通过用程序指令控制单片机引脚输出高电平或低电平。例如： 指令CLR是清零的意思，CLR P1.0的意思就是让单片机的P1.0端口输出低电平；而指令SETB是置1的意思，SETB P1.0的意思就是让单片机P1.0端口输出高电平[5]。

P2口是一个带内部上拉电阻的8位双向I/O口，而且P2口具有驱动4个LSTTL负载的能力。P2端口置1时，内部上拉电阻将端口的电位拉到高电平，作为输入口使用；在对内部的Flash程序存储器编程时，P2口接收高8位地址和控制信息，而在访问外部程序和16位外部数据存储器时，P2口就送出高8位地址。在访问8位地址的外部数据存储器时，P2引脚上的内容在此期间不会改变。

P3口也是一个带内部上拉电阻的8位双向I/O口，P3口能驱动4个LSTTL负载，这8个引脚还用于专门的第二功能[5]。P3口作为通用I/O口接口时，第二功能输出线为高电平。P3口置1时，内部上拉电阻将端口电位拉到高电平，作输入口使用；在对内部Flash程序存储器编程时，此端接控制信息。P3口的第二功能，如表4-2所示。

表3-2 P3口第二功能表

|  |  |
| --- | --- |
| P3引脚 | 兼用功能 |
| P3.0 | 串行通讯输入口（RXD） |
| P3.1 | 串行通讯输出口（TXD） |
| P3.2 | 外部中断0请求输入端（ INT0） |
| P3.3 | 外部中断1请求输入端（INT1） |
| P3.4 | 定时器0输入端(T0) |
| P3.5 | 定时器1输入端(T1) |
| P3.6 | 外部数据存储器写选通信号输出端（/WR） |
| P3.7 | 外部数据存储器写选通信号输出端（/RD） |

（5）其它控制或复用引脚

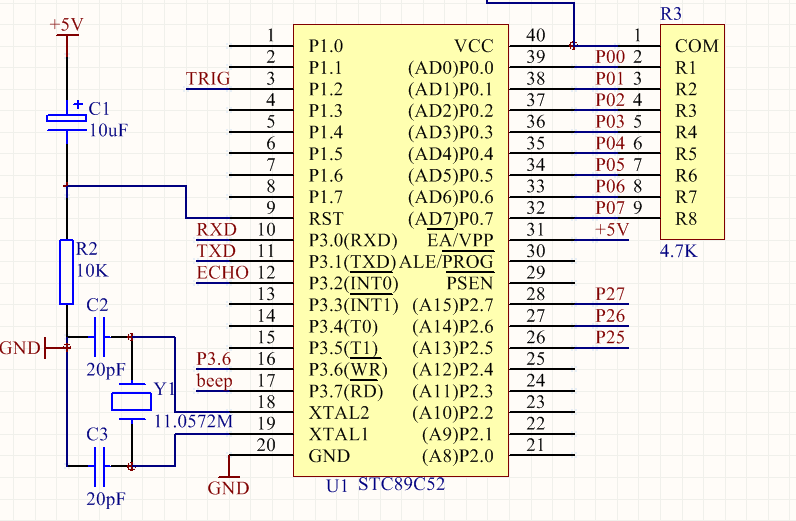
（a）ALE/PROG（30脚）：地址锁存有效信号输出端。在访问片外存储器时，ALE（地址锁存允许）以每机器周期两次进行信号输出，其下降沿用于控制锁存P0口输出的低8位地址；在不访问片外存储器的时候，ALE端仍以不变的频率输出脉冲信号(此频率是振荡器频率的1/6),而在访问片外数据存储器时，ALE脉冲会跳空一个,此时是不可以做为时钟输出[5]。对片内含有EPROM的机型在编程时，这个引脚用于输入编程脉冲/PROG的输入端[5]。

（b）/PSEN（29脚）：片外程序存储器读选通信号输出端，低电平时有效。当89S51从外部程序存储器取指令或常数时，每个机器周期内输出2个脉冲即两次有效，以通过数据总线P0口读回指令或常数。但在访问片外数据存储器时，/PSEN将不会有脉冲输出[5]。

（c）/EA/Vpp（31脚）：/EA为片外程序存储器访选用端。当该引脚访问片外程序存储器时，应该输入的是低电平，要使89S51只访问片外程序存储器,这时该引脚必须保持低电平；而在对Flash存储器编程时，用于施加Vpp编程电压。

3.1.2 单片机最小系统

单片机最小系统是其他拓展系统的最基本的基础，单片机最小系统是指一个真正可用的单片机最小配置系统即单片机能工作的系统。对于STC89C52RC单

图3-4 单片机最小系统原理图

片机，由于片内已经自带有了程序存储器，所以只要单片机外接时钟电路和复位电路就可以组成了单片机的最小系统了。单片机的最小系统如图3.4所示。

3.2显示电路的设计

具体的电路如图4-8

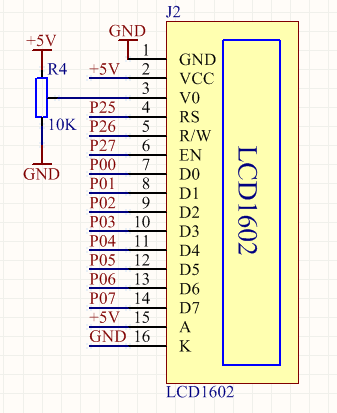


图4-8显示电路的设计

3.2.1 液晶显示器的优点

本系统显示电路选择液晶显示器LCD1602，在单片机系统中应用晶液显示器作为输出器件有以下几个优点：

① 显示质量高

由于液晶显示器每一个点在收到信号后就一直保持那种色彩和亮度，恒定发光，而不像阴极射线管显示器（CRT）那样需要不断刷新新亮点。因此，液晶显示器画质高且不会闪烁。

② 数字式接口

液晶显示器都是数字式的，和单片机系统的接口更加简单可靠，操作更加方便。

③ 体积小、重量轻

液晶显示器通过显示屏上的电极控制液晶分子状态来达到显示的目的，在重量上比相同显示面积的传统显示器要轻得多。

④ 功耗低

相对而言，液晶显示器的功耗主要消耗在其内部的电极和驱动IC上，因而耗电量比其它显示器要少得多。

### 3.2.2 液晶显示器的原理

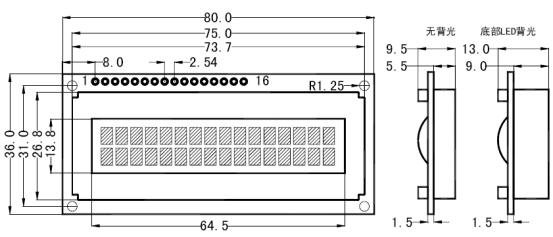
液晶显示的原理是利用液晶的物理特性，通过电压对其显示区域进行控制，有电就有显示，这样即可以显示出图形。液晶显示器具有厚度薄、适用于大规模集成电路直接驱动、易于实现全彩色显示的特点，目前已经被广泛应用在便携式电脑、数字摄像机、PDA移动通信工具等众多领域。液晶显示的分类方法有很多种，通常可按其显示方式分为段式、字符式、点阵式等。除了黑白显示外，液晶显示器还有多灰度有彩色显示等。如果根据驱动方式来分，可以分为静态驱动（Static）、单纯矩阵驱动（Simple Matrix）和主动矩阵驱动（Active Matrix）三种。

点阵图形式液晶由M×N个显示单元组成，假设LCD显示屏有64行，每行有128列，每8列对应1字节的8位，即每行由16字节，共16×8=128个点组成，屏上64×16个显示单元与显示RAM区1024字节相对应，每一字节的内容和显示屏上相应位置的亮暗对应。例如屏的第一行的亮暗由RAM区的000H——00FH的16字节的内容决定，当（000H）=FFH时，则屏幕的左上角显示一条短亮线，长度为8个点；当（3FFH）=FFH时，则屏幕的右下角显示一条短亮线；当（000H）=FFH，（001H）=00H，（002H）=00H，……（00EH）=00H，（00FH）=00H时，则在屏幕的顶部显示一条由8段亮线和8条暗线组成的虚线。这就是LCD显示的基本原理。

用LCD显示一个字符时比较复杂，因为一个字符由6×8或8×8点阵组成，既要找到和显示屏幕上某几个位置对应的显示RAM区的8字节，还要使每字节的不同位为“1”，其它的为“0”，为“1”的点亮，为“0”的不亮。这样一来就组成某个字符。但由于内带字符发生器的控制器来说，显示字符就比较简单了，可以让控制器工作在文本方式，根据在LCD上开始显示的行列号及每行的列数找出显示RAM对应的地址，设立光标，在此送上该字符对应的代码即可。

汉字的显示一般采用图形的方式，事先从微机中提取要显示的汉字的点阵码（一般用字模提取软件），每个汉字占32B，分左右两半，各占16B，左边为1、3、5……右边为2、4、6……根据在LCD上开始显示的行列号及每行的列数可找出显示RAM对应的地址，设立光标，送上要显示的汉字的第一字节，光标位置加1，送第二个字节，换行按列对齐，送第三个字节……直到32B显示完就可以LCD上得到一个完整汉字。

LCD1602分为带背光和不带背光两种，基控制器大部分为HD44780，带背光的比不带背光的厚，是否带背光在应用中并无差别，两者尺寸差别如下图3.7所示。

图4-9 1602LCD尺寸图

3.3报警电路的设计

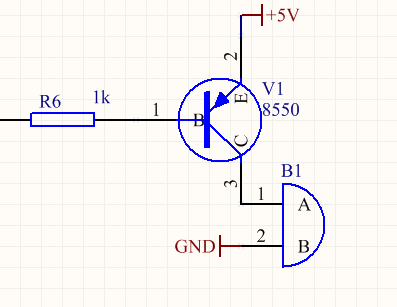
图所示为蜂鸣器报警电路。由于单片机管脚的灌电流比拉电流容量大，因此电路设计为低电平输出时蜂鸣器响，高电平关闭。当P3.7脚输出低电平时，PNP型三极管8550导通，有集电极电流通过，蜂鸣器鸣响。当P3.7脚输出高电平时，三极管截止，蜂鸣器关闭。

图3 报警电路的设计

3.4 温度采集电路设计

本部分电路主要通过传感器来实现对温度的测量，本系统选用的是DS18B20传感器，DS18B20是美国DALLAS公司继DS1820之后推出的增强型单总线数字温度传感器。它在测温精度、转换时间、传输距离、分辨率等方面较DS1820有了很大的改进，给用户带来了更方便的使用和更令人满意的效果。

3.4.1 DS18B20的介绍

1.DS18B20的结构

DS18B20主要由寄生电源、温度传感器、64位串行ROM单线接口、存储中间数据的高速暂存器（内含便笺式RAM）、用于存储用户设定的温度上下限值的TH和TL触发器存储与控制逻辑、8位循环冗余检验码(CRC)发生器部分。

DS18B20管脚排列如图3.5所示：

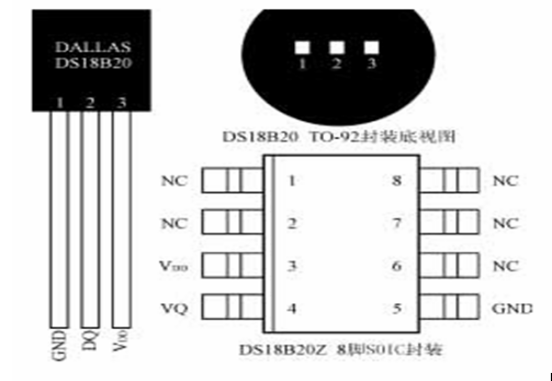


图3.5 DS18B20的引脚图

本设计使用的是三引脚的产品。其中，1号引脚接地，2号引脚接数据端，3号引脚接电源。

2.DS18B20的主要特点

数字型智能温度传感器有以下主要特点：

(1) 适应电压范围更宽，电压范围：3.0～5.5V，在寄生电源方式下可由数据线供电。

(2) 独特的单线接口方式，DS18B20 在与微处理器连接时仅需要一条口线即可实现微处理器与 DS18B20 的双向通讯。

(3) DS18B20 支持多点组网功能，多个 DS18B20 可以并联在唯一的单总线上，实现组网多点测温 。

(4) DS18B20 在使用中不需要任何外围元件，全部传感元件及转换电路集成在形如一只三极管的集成电路内 。

(5) 温范围－55℃～＋125℃，在-10～+85℃时精度为±0.5℃ 。

(6) 温度分辨力可编程。DS18B20的数字温度输出可进行9～12位编程。

在实际应用时，需要在分辨力与转换时间两者之间权衡考虑。当DS18B20工作在12位分辨力时，温度与数字输出的对应关系见表3.1。

表3.1 DS18B20输出数据与温度的对应关系

|  |  |  |
| --- | --- | --- |
| 温度/℃ | 数字输出（二进制） | 数字输出（十六进制） |
| +125 | 0000 0111 1101 0000 | 07D0H |
| +85 | 0000 0101 0101 0000 | 0550H |
| +25.0625 | 0000 0001 1001 0001 | 0191H |
| +10.125 | 0000 0001 1010 0010 | 00A2H |
| +0.5 | 0000 0001 1010 1000 | 0008H |
| 0 | 0000 0000 0000 0000 | 0000H |
| -10.125 | 1111 1111 0101 1110 | FF5EH |
| -25.0625 | 1111 1110 0101 1111 | FE6FH |
| -55 | 1111 1100 1001 0000 | FC90H |

(7) 测量结果直接输出数字温度信号，以“一线总线”串行传送给 CPU，同时可传送 CRC 校验码，具有极强的抗干扰纠错能力 。

(8) 测量结果直接输出数字温度信号，以“一线总线”串行传送给 CPU，同时可传送 CRC 校验码，具有极强的抗干扰纠错能力 。

3.DS18B20测温原理

用一个高温度系数的振荡器确定一个门周期，内部计数器在这个门周期内对一个低温度系数的振荡器的脉冲进行计数来得到温度值。计数器被预置到对应于-55℃的一个值。如果计数器在门周期结束前达到0，则温度寄存器(同样被预置到-55℃)的值增加，表明所测量的温度大于-55℃。

同时，计数器被复位到一个值，这个值由斜坡式累加器电路确定，斜坡式累加器电路用来补偿感温振荡器的抛物线特性。然后计数器又开始计数直到0，如果门周期仍未结束，将重复这一过程。

斜坡式累加器用来补偿感温振荡器的非线性，以期在测温时获得比较高的分辨力。这是通过改变计数器对温度每增加一度所需计数的值来实现的。因此，要想获得所需的分辨力，必须同时知道在给定温度下计数器得值和每一度的计数值。温度测电路的方框图如图3.6所示：

停止

增加

斜率累加器

减法计数器1

减法计数器2

比较计数器

温度寄存器

减到0

减到0

预置

高温度系数振荡器

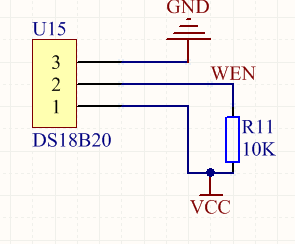
低温度系数振荡器

预置

图DS18B20测温原理图

3.4.2 测温电路设计

本文中测温电路主要使用DS18B20传感器，通过P1.0口与单片机相连，实现数据的传递，



其具体硬件原理图如图3.7所示：

图3.7 测温电路原理图

DS18B20芯片有两种供电方式：寄生电源供电方式和外部电源供电方式。本文采用的是外部电源供电方式。

在外部电源供电方式下，DS18B20工作电源由VDD引脚接入，此时I/O线不需要强上拉，不存在电源电流不足的问题，可以保证转换精度，同时在理论上总线上可以挂接任意多个DS18B20传感器，组成多点测温系统。注意：在外部供电的方式下，DS18B20的GND引脚不能悬空，否则不能转换温度，读取的温度总是85℃。外部电源供电方式是DS18B20的最佳工作方式，工作稳定可靠，抗干扰能力强，而且电路也比较简单，可以开发出稳定可靠的多点温度监控系统，在外接电源方式下，可以充分发挥DS18B20宽电源电压范围的优点，即使电源电压VCC降到3V，依然能够保证温度量精度。

在实际应用中还需要注意的是，连接DS18B20的总线电缆是有长度限制的，试验中，当采用普通信号电缆传输长度超过50m时，读取的温度数据将发生错误，当将总线电缆改为双绞线带屏蔽电缆时，正常通信距离可达150m，当采用每米绞合次数更多的双绞线带屏蔽电缆时，正常通信距离进一步加长，这种情况主要是由总线分布电容使信号波形发生畸形造成的，因此，在用DS18B20进行长距离测温系统设计时，要充分考虑总线分布电容和阻抗匹配问题。

3.5继电器控制电路设计

该部分通过单片机的P口输出的高低电平来控制固态继电器的通断，从而决定电热丝是否加热，当P口输出低电平时，加热电阻通电，周围的温度缓慢升高，DS18B20测得的温度值也升高；当P口输出高电平时，加热电路断开，温度回落。



4 系统软件的设计

根据温度控制系统硬件设计的软件程序，实际上就是对系统的管理程序和控制程序进行设计。管理程序是对显示LED进行动态刷新、控制指示灯、处理键盘的扫描和响应等。控制程序是对被控制过程进行采样、数据处理，根据控制算式进行计算和输出等。由于整个系统软件相当庞大，为便于编写、调试和修改，系统软件的编制采用了模块化结构，及整个控制软件由许多独立的小模块组成，他们之间通过软件接口连接，主控程序主要包括条件判断和子程序调用等关键部分。在进行微机控制系统设计时，系统设计人员必须把系统要执行的任务和应具备的功能合理地分配给硬件和软件来实现，既要考虑系统的价格，又要考虑系统满足实时性要求的工作进度，做到硬件、软件合理权衡，并尽量节省机器时间和内存空间。

4.1系统的总体流程图

开 始

初始化DS18B20

显示当前温度

判断当前温度值

超过设定

温度上限

启动风扇

降低温度

红灯亮

设定温度上、下限

启动电热炉升高温度

是

否

低于设定

温度下限

是

绿灯亮

否

4.2 测温程序流程图

N

Y

返回

读取温度值

存储操作命令

ROM操作命令

开始

初始化

DS18B20是否存在

4.3 控制部分

“K”

“G”

接收数据

P0.1=0加热丝工作

P0.1=1加热丝停止工作

判断数据为“K”还是“G”

开始

结论

通过本次温度监控系统的设计,我大有收获，在制作过程中，一定要注意的每个工作步骤的检查，确保制作成功。比如在合理布线，检查装配无误的情况下，如果还出现电路无输出的情况，那么可以肯定是原理图错误，这时就要回到原理图进行检查。总体的检查顺序应该是原理图、PCB图、装配情况、焊接工艺。从整体来说这是一个复杂的过程，要细心谨慎，沉着冷静，反复检查，直到找到原因为止。

这次毕业设计历时至少3个月，从一开始的确定课题，到后来的资料查找、理论学习，再有就是近来的调试和测试过程，这一切都使我的理论知识和动手能力进一步得到频率合成电路课题中包含了通信电路和单片机部分知识，可以说是对通信电路知识的一次全面综合。在画原理图、PCB布线、安装和调试过程中不可避免地遇到各种问题，这要求保持沉着冷静，联系书本理论知识积极地思考，实在解决不了可以请教同学或指导老师。虽然在制作过程中不可避免地遇到很多问题，但是最后还是在老师以及同学的帮助下圆满解决了这些问题，实现了整个系统设计与最后调试，相关指标达到期望的要求，很好地完成了本次设计任务。

经过三年学习的积累，在已经掌握相关专业方面知识及其它各方面知识的情况下，我认真严肃的完成了我的毕业设计。

从得到题目到查找资料，从对题目的研究设定到PCB电路板的制作，从电路板的调试到失败后再一次全部重新开始……在这一个充满挑战伴随挫折，充满热情伴随打击的过程中，我感触颇深，它已不仅是一个对我四年学习知识情况和我的应用动手能力的检验，而且还是对我的钻研精神，面对困难的心态，做事的毅力和耐心的考验。我在这个过程中深刻的感受到了做毕业设计的意义所在，和我一样真正投入了身心去做的人也一定会有同样的感触。

参考文献

[1] 何立民.单片机应用系统设计系统配置与接口技术[M].北京：北京航空航天大学，1990.

[2] 李晓荃.单片机原理与应用[M]. 北京:电子工业出版社，2000.

[3] 刘和平.单片机原理及应用[M].重庆：重庆大学出版社，2002 .

[4] 徐爱钧.单片机高级语言 C51 应用程序设计[M]. 北京：电子工业出版社，2002.

[5] 谢自美.电子线路设计.实验.测试(第二版) [M].武汉：华中科技大学出版社，2000.

[6] 江国强.现代数字逻辑电路.北京：电子工业出版社，2002 .

[7] 张勇.PROTEL 99SE 电路设计技术入门与应用(第一版).北京：电子工业出版社，2002 .

[8] 樊昌信.通信原理(第五版)[M].北京：国防工业出版社,2001 .

[9] Richard c.Dorf.modern conctrol systerm[M].BEIJING:Science Publishing House，2002.

[10] Donald A. Neamen. Electronic circuit analysis and design[M].Tsinghua University Press and Springer Verlag.2002.

致谢

在本次毕业设计中，我得到了指导老师方娜的热心指导。自始至终关心督促毕业设计进程和进度。帮助解决毕业设计中遇到的许多问题。还不断向我们传授分析问题和解决问题的办法，并指出了正确的努力方向，使我在毕设过程中少走很多弯路。同时，她还提供给我们专门的各种设备及场所，在调试过程中能够有充足的时间。在这里非常感谢方娜老师的指导和帮助，并致以诚挚的谢意！

同时，身边的同学给了我许多的帮助。在此，我向身边关心我的同学致以诚挚的谢意！另外，系里的领导和老师也给了我们必要的指导，我也向系和年级的领导们表示衷心的感谢！最后感谢学院对我这几年的培养。

附录 主要源程序

#include < reg51.h >

#include < intrins.h >

#define uchar unsigned char

#define uint unsigned int

sbit DQ = P3^4 ; //定义DS18B20端口DQ

bit presence ; //检测18b20是否插好

sbit LCD\_RS = P1^0 ;

sbit LCD\_RW = P1^1 ;

sbit LCD\_EN = P1^2 ;

sbit s1 = P1^3;

sbit s2 = P1^4;

sbit s3 = P1^5;

sbit s4 = P1^6;

sbit LED = P2^0;

sbit BEEP = P2^1;

sbit KM1 = P3^6;

sbit KM2 = P3^7;

uint i;

uchar code cdis1[ ] = {"T-H: C T-L: C"};

uchar code cdis2[ ] = {" WENDU: . C "} ;

uchar code cdis3[ ] = {"DS18B20 ERR0R "} ;

uchar code cdis4[ ] = {" PLEASE CHECK "} ;

char MH=60,ML=20;

unsigned char data temp\_data[2] = {0x00,0x00} ;

unsigned char data display[5] = {0x00,0x00,0x00,0x00,0x00} ;

unsigned char code ditab[16] = {0x00,0x01,0x01,0x02,0x03,0x03,0x04,0x04,

0x05,0x06,0x06,0x07,0x08,0x08,0x09,0x09} ;

void beep() ;

unsigned char code mytab[8] = {0x0C,0x12,0x12,0x0C,0x00,0x00,0x00,0x00} ;

#define delayNOP() ; {\_nop\_() ;\_nop\_() ;\_nop\_() ;\_nop\_() ;} ;

void keyscan();

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void delay1(int ms)

{

unsigned char y ;

while(ms--)

{

for(y = 0 ; y<250 ; y++)

{

\_nop\_() ;

\_nop\_() ;

\_nop\_() ;

\_nop\_() ;

}

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*检查LCD忙状态 \*/

/\*lcd\_busy为1时，忙，等待。lcd-busy为0时,闲，可写指令与数据。 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

bit lcd\_busy()

{

bit result ;

LCD\_RS = 0 ;

LCD\_RW = 1 ;

LCD\_EN = 1 ;

delayNOP() ;

result = (bit)(P0&0x80) ;

LCD\_EN = 0 ;

return(result) ;

}

/\*写指令数据到LCD \*/

/\*RS=L，RW=L，E=高脉冲，D0-D7=指令码。 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void lcd\_wcmd(uchar cmd)

{

while(lcd\_busy()) ;

LCD\_RS = 0 ;

LCD\_RW = 0 ;

LCD\_EN = 0 ;

\_nop\_() ;

\_nop\_() ;

P0 = cmd ;

delayNOP() ;

LCD\_EN = 1 ;

delayNOP() ;

LCD\_EN = 0 ;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*写显示数据到LCD \*/

/\*RS=H，RW=L，E=高脉冲，D0-D7=数据。 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void lcd\_wdat(uchar dat)

{

while(lcd\_busy()) ;

LCD\_RS = 1 ;

LCD\_RW = 0 ;

LCD\_EN = 0 ;

P0 = dat ;

delayNOP() ;

LCD\_EN = 1 ;

delayNOP() ;

LCD\_EN = 0 ;

}

/\* LCD初始化设定 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void lcd\_init()

{

delay1(15) ;

lcd\_wcmd(0x01) ; //清除LCD的显示内容

lcd\_wcmd(0x38) ; //16\*2显示，5\*7点阵，8位数据

delay1(5) ;

lcd\_wcmd(0x38) ;

delay1(5) ;

lcd\_wcmd(0x38) ;

delay1(5) ;

lcd\_wcmd(0x0c) ; //显示开，关光标

delay1(5) ;

lcd\_wcmd(0x06) ; //移动光标

delay1(5) ;

lcd\_wcmd(0x01) ; //清除LCD的显示内容

delay1(5) ;

}

/\* 设定显示位置 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void lcd\_pos(uchar pos)

{

lcd\_wcmd(pos | 0x80) ; //数据指针=80+地址变量

}

/\*自定义字符写入CGRAM \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void writetab()

{

unsigned char i ;

lcd\_wcmd(0x40) ; //写CGRAM

for (i = 0 ; i< 8 ; i++)

lcd\_wdat(mytab[ i ]) ;

}

/\*us级延时函数 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Delay(unsigned int num)

{

while( --num ) ;

}

/\*初始化ds1820 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Init\_DS18B20(void)

{

DQ = 1 ; //DQ复位

Delay(8) ; //稍做延时

DQ = 0 ; //单片机将DQ拉低

Delay(90) ; //精确延时 大于 480us

DQ = 1 ; //拉高总线

Delay(8) ;

presence = DQ ; //如果=0则初始化成功 =1则初始化失败

Delay(100) ;

DQ = 1 ;

return(presence) ; //返回信号，0=presence,1= no presence

}

/\* 读一个字节 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

ReadOneChar(void)

{

unsigned char i = 0 ;

unsigned char dat = 0 ;

for (i = 8 ; i > 0 ; i--)

{

DQ = 0 ; // 给脉冲信号

dat >>= 1 ;

DQ = 1 ; // 给脉冲信号

if(DQ)

dat |= 0x80 ;

Delay(4) ;

}

return (dat) ;

}

/\* 写一个字节 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

WriteOneChar(unsigned char dat)

{

unsigned char i = 0 ;

for (i = 8 ; i > 0 ; i--)

{

DQ = 0 ;

DQ = dat&0x01 ;

Delay(5) ;

DQ = 1 ;

dat>>=1 ;

}

}

/\* 读取温度 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Read\_Temperature(void)

{

Init\_DS18B20() ;

WriteOneChar(0xCC) ; // 跳过读序号列号的操作

WriteOneChar(0x44) ; // 启动温度转换

Init\_DS18B20() ;

WriteOneChar(0xCC) ; //跳过读序号列号的操作

WriteOneChar(0xBE) ; //读取温度寄存器

temp\_data[0] = ReadOneChar() ; //温度低8位

temp\_data[1] = ReadOneChar() ; //温度高8位

}

/\* 数据转换与温度显示 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Disp\_Temperature()

{

display[4]=temp\_data[0]&0x0f ;

display[0]=ditab[display[4]]+0x30 ;//查表得小数位的值

display[4]=((temp\_data[0]&0xf0)>>4)|((temp\_data[1]&0x0f)<<4) ;

display[3]=display[4]/100+0x30 ;

display[1]=display[4]%100 ;

display[2]=display[1]/10+0x30 ;

display[1]=display[1]%10+0x30 ;

if(display[3]==0x30) //高位为0，不显示

{

display[3]=0x20 ;

if(display[2]==0x30) //次高位为0，不显示

display[2]=0x20 ;

}

lcd\_pos(0x48) ;

lcd\_wdat(display[3]) ; //百位数显示

lcd\_pos(0x49) ;

lcd\_wdat(display[2]) ; //十位数显示

lcd\_pos(0x4a) ;

lcd\_wdat(display[1]) ; //个位数显示

lcd\_pos(0x4c) ;

lcd\_wdat(display[0]) ; //小数位数显示

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 蜂鸣器响一声 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void beep()

{

unsigned char y ;

for (y=0 ;y<100 ;y++)

{

Delay(60) ;

BEEP=!BEEP ; //BEEP取反

}

BEEP=1 ; //关闭蜂鸣器

Delay(40000) ;

}

/\* DS18B20 OK 显示菜单 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Ok\_Menu ()

{

uchar m ;

lcd\_init() ; //初始化LCD

lcd\_pos(0) ; //设置显示位置为第一行的第1个字符

m = 0 ;

while(cdis1[m] != '\0')

{ //显示字符

lcd\_wdat(cdis1[m]) ;

m++ ;

}

lcd\_pos(0x40) ; //设置显示位置为第二行第1个字符

m = 0 ;

while(cdis2[m] != '\0')

{

lcd\_wdat(cdis2[m]) ; //显示字符

m++ ;

}

writetab() ; //自定义字符写入CGRAM

delay1(5) ;

lcd\_pos(0x4d) ;

lcd\_wdat(0x00) ; //显示自定义字符

}

/\* DS18B20 ERROR 显示菜单 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Error\_Menu ()

{

uchar m ;

lcd\_init() ; //初始化LCD

lcd\_pos(0) ; //设置显示位置为第一行的第1个字符

m = 0 ;

while(cdis3[m] != '\0')

{ //显示字符

lcd\_wdat(cdis3[m]) ;

m++ ;

}

lcd\_pos(0x40) ; //设置显示位置为第二行第1个字符

m = 0 ;

while(cdis4[m] != '\0')

{

lcd\_wdat(cdis4[m]) ; //显示字符

m++ ;

}

}

void pross()

{

if(display[4]>MH||display[4]<ML)

{LED=0;

BEEP=0;

} else

{

LED=1;

BEEP=1;

}

if(display[4]>MH)

{

KM1=0;

} else

{

KM1=1;

}

if(display[4]<ML)

{

KM2=0;

} else

{

KM2=1;

}

}

void T1\_Init()

{

// TMOD|= 0x11;

//TH1 = 0xD8; //10

// TL1 = 0xF0;

// EA=1;

// ET1=1;

// TR1=1;

// IP=0x04;

// EX0=1;//以上为定时器，外中断，串口中断的设置

}

void T1zd() interrupt 3

{

TH1 = 0xD8;

TL1 = 0xF0;

i++;

if(i==1)

{

i=0;

}

}

void juged()

{

lcd\_pos(0x04);

lcd\_wdat(MH/10+48);

lcd\_pos(0x05);

lcd\_wdat(MH%10+48);

lcd\_pos(0x0d);

lcd\_wdat(ML/10+48);

lcd\_pos(0x0e);

lcd\_wdat(ML%10+48);

keyscan();

pross();

}

/\* 主函数 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

KM1=0;

KM2=0;

while(1)

{ Ok\_Menu () ;

do

{

Read\_Temperature() ; //读取温度

Disp\_Temperature() ; //显示温度

juged();

}

while(!presence) ;

Error\_Menu () ;

do

{

Init\_DS18B20() ;

beep() ;

}

while(presence) ; }

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*键值扫描函数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void keyscan()

{ uchar j,i;

if(s1==0)

{

for(j=0;j<60;j++);

if(s1==0)

{

while(!s1);

MH++;

if(MH>99)

MH=99;

}

}

if( s2==0)

{

for(j=0;j<60;j++);

if( s2==0)

{

while(!s2);

MH--;

if(MH<0)

MH=0;

}

}

if( s3==0)

{

for(j=0;j<60;j++);

if( s3==0)

{

while(!s3);

ML++;

if(ML>99)

ML=99;

}

}

if( s4==0)

{

for(j=0;j<60;j++);

if( s4==0)

{

while(!s4);

ML--;

if(ML<0)

ML=0;

}

}

}