

---

## 摘要

20 世纪末，随着计算机科学的发展，计算机已深入地渗透到我们的生活中，要想学好温度测控，只有扎扎实实的把单片机知识学好，因为温度测控是基于单片机知识的。今天，我们的生活环境和工作环境有越来越多称之为单片机的小电脑在为我们服务，单片机在工业控制，尖端武器，通信设备，信息处理，家用电器等各测，控领域的应用中独占鳌头。时下，家用电器和办公设备的智能化，遥控化，模糊控制化已成为世界潮流，而这些高性能无一不是靠单片机来实现的，同时在使用单片机的前提下我们也还要用到 A/D 数模转换。我们作为 21 世纪的大学生必须具备单片机知识。现在随着社会对人才素质要求的不断提高，我们也要不断的充实自己，以适应社会的发展。

**关键词：** 单片机    DS18B20    传感器

## Abstract

20 century's ends, along with the computer science development, the computer thoroughly seeped to in ours life, must want to learn the temperature observati on and control, only then is solid learns the monolithic integrated circuit knowledge, because the temperature observati on and control is based on he monolithic integrated circuit knowledge. Today, our living conditions and the working conditions have more and more called it in serves the

monolithic integrated circuit small computer for us, the monolithic integrated circuit in the industry control, the state-of-art weapon, the communication facility, the information processing, the domestic electric appliances and so on measured respectively, controls the domain in the application to lead the field. Present, domestic electric appliances and office equipment intellectualization, remote control, the fuzzy control has become the trends in the world, but these high performances do not have one are not depend on, at the same time in uses the monolithic integrated circuit which the monolithic integrated circuit realizes under the premise we also to have to use to A/D digital-analog conversion. We took 21st century the university student must have the monolithic integrated circuit knowledge. Now along with the society to the talented person quality request unceasing enhancement, we also wants unceasingly to enrich oneself, by adapts society's development.

**Key word:** MCU DS18B20 sensor

## 目 录

前 言.....	4
功能说明.....	5
一 总体设计方案.....	6
二 电路原理图.....	7
三 MCS-51 系列单片机简介.....	8
四 温度传感器 DS1820 简介.....	18
五 软件编程.....	24
致 谢.....	34
心得体会.....	35
参考文献.....	36

## 前言

随着人们生活水平的不断提高,单片机控制无疑是人们追求的目标之一,它所给人带来的方便也是不可否定的,。温度测控技术也在各个领域应用越来越广泛,同时温度测量也被人们所异常关注,一时间涌现出大量各式各样的测试温度仪器,这些仪器的原理到底是怎样的呢?说到底也是属于温度测控的范畴,其中数字温度计就是一个典型的例子,但人们对它的要求越来越高,要为现代人工作、科研、生活、提供更好的更方便的设施就需要从数单片机技术入手,一切向着数字化控制,智能化控制方向发展。

本设计所介绍的数字温度计与传统的温度计相比,具有读数方便,测温范围广,测温准确,其输出温度采用数字显示,主要用于对测温比较准确的场所,或科研实验室使用,测温传感器使用 DS18B20,用 3 位共阳极 LED 数码管以串口传送数据,实现温度显示,能准确达到以上要求。

## 功能说明

(1) 将 AD590 作为室温传感器，当温度变化时，AD590 会产生电流的变化，经 OPA1 将电流转换为电压，由 OPA2 作为零为调整，最后由 OPA3 反相放大 10 倍。

(2) ADC0804 输出最大转换值=FFH (255)。OPA3 为放大 10 倍时。则本电路最大测量温度为；最大显示温度为  $5.1/10V=0.51V$ ，即  $51^{\circ}C$  (10 为放大倍数)。  $255X=51$ ，知  $X=0.2$ ，即先乘 2 再除 10。  $FF \rightarrow 255 \rightarrow 255 \times 2 \rightarrow 510$ ，  $R4=0.5 R3=10$ 。即  $D4=0$ ，  $D3=5$ ，  $D2=1$ ，  $D1=0$ ，本电路显示器只取  $D3$ 、 $D2$  两位数。

(3) 按下 P2.1 按钮，放开后立即进入温度设定模式，显示设定最高温度为  $34^{\circ}C$  (建立在 TABLE 内) 每按一次设定温度将减少  $1^{\circ}C$ ，直至最低温度  $20^{\circ}C$ ，再按一次回到  $34^{\circ}C$ 。

(4) 当室温高于设定温度，压缩机 (P3.0) 运转，使室温降低，当室温低于设定值时，压缩机停止运行。

(5) 当进入设定温度模式，如未按下设定按钮 (P2.1) 经数秒后自动解除设定模式，回到室温显示模式。

(6) 本程序以计时中断，每 50ms 中断一次，比较室温一次，而令压缩机运转和停止。

## 一 总体设计方案

### 1.1 数字温度计设计方案论证

考虑到用温度传感器，在单片机电路设计中，大多都是使用传感器，所以这是非常容易想到的，所以可以采用一只温度传感器 DS18B20，此传感器，可以很容易直接读取被测温度值，进行转换，就可以满足设计要求。

### 1.2 方案的总体设计框图

温度计电路设计总体设计方框图如图 1 所示，控制器采用单片机 AT89S51，温度传感器采用 DS18B20，用 3 位 LED 数码管以串口传送数据实现温度显示。

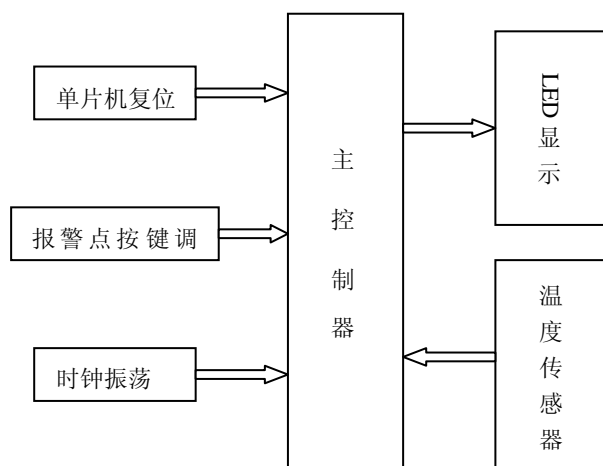
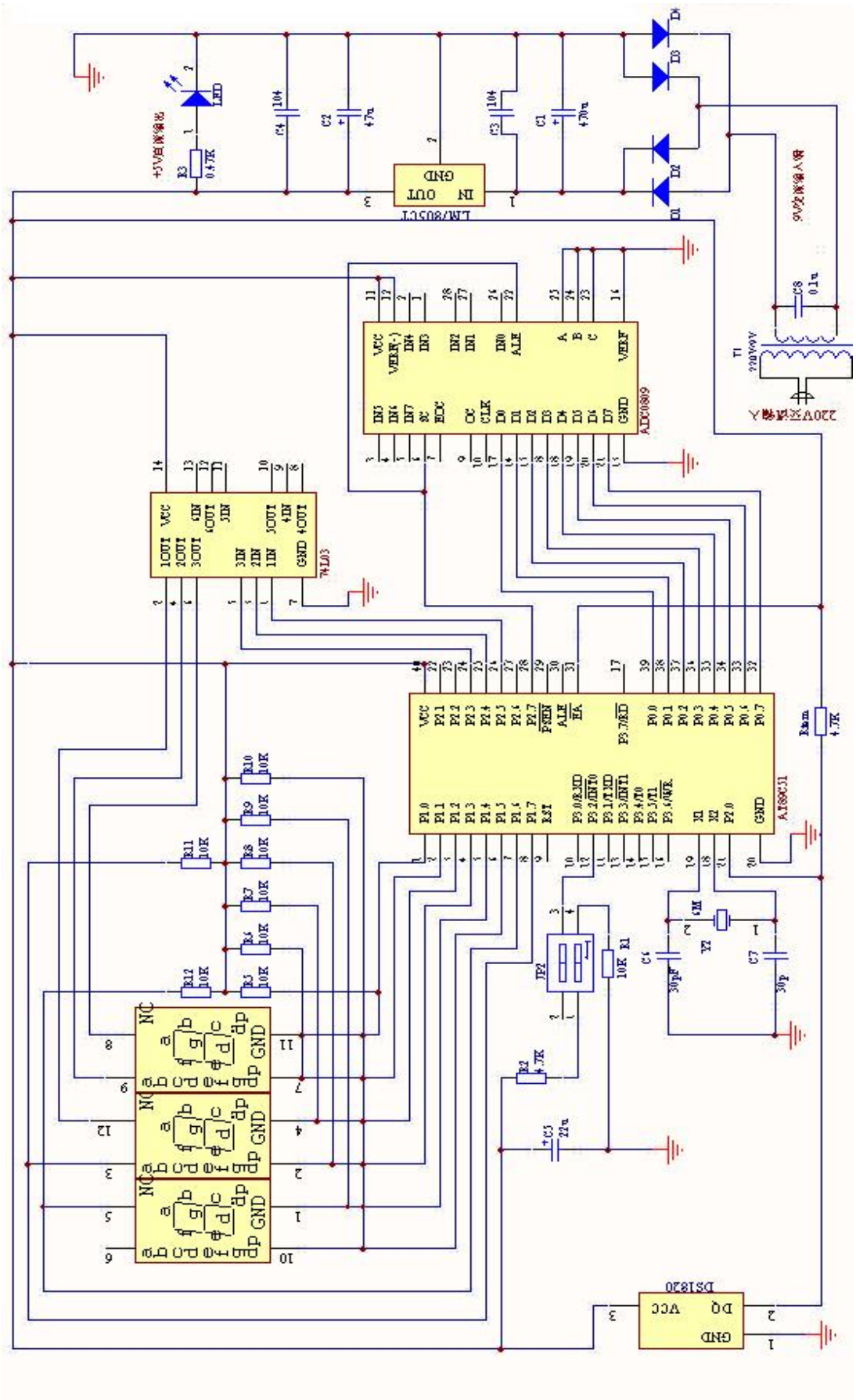


图 1 总体设计方框图

## 二 电路原理图



### 三 MCS-51 系列单片机简介

MCS-51 系列单片机研制于 1980 年，由 Intel 公司所开发，其结构是 8048 的延伸，改进了 8048 的缺点，其 ROM、RAM 都可扩充至 64KB，也增添了如乘 (MUL)、除 (DIV)、减 (SUBB)、比较 (CJNE)、栈入 (PUSH)、栈出 (POP)、16 位数据指针、布尔代数运算等指令，以及串行通信能力和 5 个中断源。8052 有 6 个中断源，MCS-51 系列单片机存储器容量如表 (二-1) 所示。MCS-51 系列单片机特点如下：

- 1、专为控制应用所设计的八位 CPU
- 2、具有布尔代数的运算能力
- 3、32 条双项且可被独立寻址的 I/O 口
- 4、芯片内有 128 字节可供存储数据的 RAM (8052: 256 字节)
- 5、内部有两组 16 位定时器 (8052 有 3 个)
- 6、具有全多工传输信号 UART
- 7、5 个中断源，且具有两级 (高/低) 优先权顺序的中断结构
- 8、芯片内有 4KB (8KB / 8052) 的程序存储器 (ROM)
- 9、芯片内有时钟 (CLOCK) 振荡器电路
- 10、程序存储器可扩展至 64KB (ROM)
- 11、数据存储器可扩展至 64KB (RAM)



表（二—1） MCS-51 系列单片机 ROM 及 RAM 容量（字节）

型号	8031	8051	8751	8032	8052	8752
ROM	0	4K	0	0	8K	0
RAM	128	128	128	256	256	256
EPROM	0	0	4K	0	0	8K

1 2、8 0 5 1 / 5 2：工厂烧写型，内含 R O M

1 3、P 8 7 5 1：一次烧写型，内含 P R O M

1 4、8 7 5 1 / 8 7 5 1：可重复烧写型，内含 E P R O M

1 5、8 7 C 5 1 / 8 7 C 5 2：省电型（低消耗功率）

## 1、时钟电路引脚

M C S - 5 1 单片机的时钟可以由内部方式和外部方式产生，X T A L 1（1 9 脚）和 X T A L 2（1 8 脚）即为单片机的两个时钟引脚。

（1）内时钟引脚 8 0 5 1 单片机片内有振荡电路，只需在 X T A L 1 和 X T A L 2 间外接石英晶体和电容组成的并联振荡电路（晶振器），晶体可以在固有频率 1.2 ~ 1.2 MHz 的晶振器之间任选晶体，电容可以在 20 ~ 60 pF 的电容之间任选，通常选择 30 pF 的瓷片电容。在单片机控制的数字显示温度计电路设计的这个部

分，就是采用内时钟引脚，其中晶振器为 6MHz，两个电容均为 30pF。

(2) 外时钟方式，X T A L 1 接地，X T A L 2 接外部振荡器。由于 X T A L 2 端的电平不是 T T L 电平，故接一个上拉电阻。外部振荡器的频率应低于 1 2 M H z。

## 2、控制信号引脚

(1) R S T / V<sub>pd</sub> (9 脚): 复位信号 / 备用电源引脚

当输入的复位信号延续 2 个机器周期以上，高电平即为有效，用以完成单片机的复位操作。复位后影响片内特殊功能寄存器的状态，但不影响片内 R A M 状态。

同一引脚的 V<sub>pd</sub> 是备用电源输入端 (V<sub>pd</sub> 接 + 5 V 备用电源)。在 V<sub>cc</sub> 断电时，为保证 R A M 中的信息不丢失，可使此引脚完成掉电保护功能。

(2) A L E / P R O G 非 (3 0 脚); 地址锁存允许信号 / 编程脉冲输入端

在系统扩展时，A L E 用于控制把 P 0 口输出的低 8 位地址送入锁存器锁存起来，以实现低位地址和数据的分时传送。此外由于 A L E 是以 1 / 6 晶振频率的固定频率输出的正脉冲，因此可作为外边时钟或外部定时脉冲使用。

对片内带有 4 kbyte E P R O M 的 8 7 5 1 编写固化程序时，P R O G 非作为编程脉冲输入端。

(3) P S E N 非 (2 9 引脚): 外边程序存储器读选通信号

为低电平有效，8 0 5 1 在访问片外程序存储器时，此引脚端输

出负脉冲作为读片外程序存储器的选通信号，以实现外部 R O M 单元的读操作。要检查 8 0 5 1 上电平后 C P U 能否正常到程序存储器中读取指令码，可以用示波器观察引脚 P S E N 非有无脉冲输出，若有说明正常。

(4) E A 非 / V<sub>pp</sub> (3 1 脚): 内部和外部程序存储器选择信号

当引脚接高电平时，C P U 只访问片内 4 kbyte 的 E P R O M / R O M，执行内部程序存储器中的指令，但在程序计数器计数超过 0 F F F 时(即地址大于 4 kbyte 时)，将自动转向执行片外大于 4 kbyte 程序存储器内的程序。

若 E A 非引脚接低电平时，C P U 只访问外部程序存储器，而不管片内是否有程序存储器。对于 8 0 3 1 单片机(片内无 R O M)需外扩 E P R O M，故必须将 E A 非引脚接地。

在对 E P R O M 编写固化程序时，需对此引脚施加 2 1 V 的编程电压。

### 3、I/O(输入/输出) 接口引脚

#### (1) 并行 I/O 接口的特点

MCS-51 有 4 个 8 位并行 I/O 接口 P0~P3，他们都是双向端口，可以进行输入或者输出操作，每个口都有口锁存器和口驱动器两部分组成。此外，它还有一个全双工串行通信口。这 4 个端口为 MCS-51 与外围器件或外围设备进行信息(数据、地址、控制信号)交换提供了多功能的输入/输出通道，也为 MCS-51 扩展外部功能、构成应用系统

提供了必要的条件。

它们的特点如下：

a、4 个并行 I/O 接口都是双向的。P0 口为漏极开路，P1、P2、P3 口均具有内部上拉电阻，它们有时被称为准双向口。

b、4 个并行口的 32 条 I/O 接口线都可以独立地用于输入或输出操作。

c、当 4 个并行口的 I/O 接口线有作输入操作时，必须对该口的锁存器进行写 1 操作，以保证从 I/O 接口线输入数据的正确性，这也是 4 个并行接口有时被称为“准”双向的含义。

## (2) I/O 接口电路功能汇总

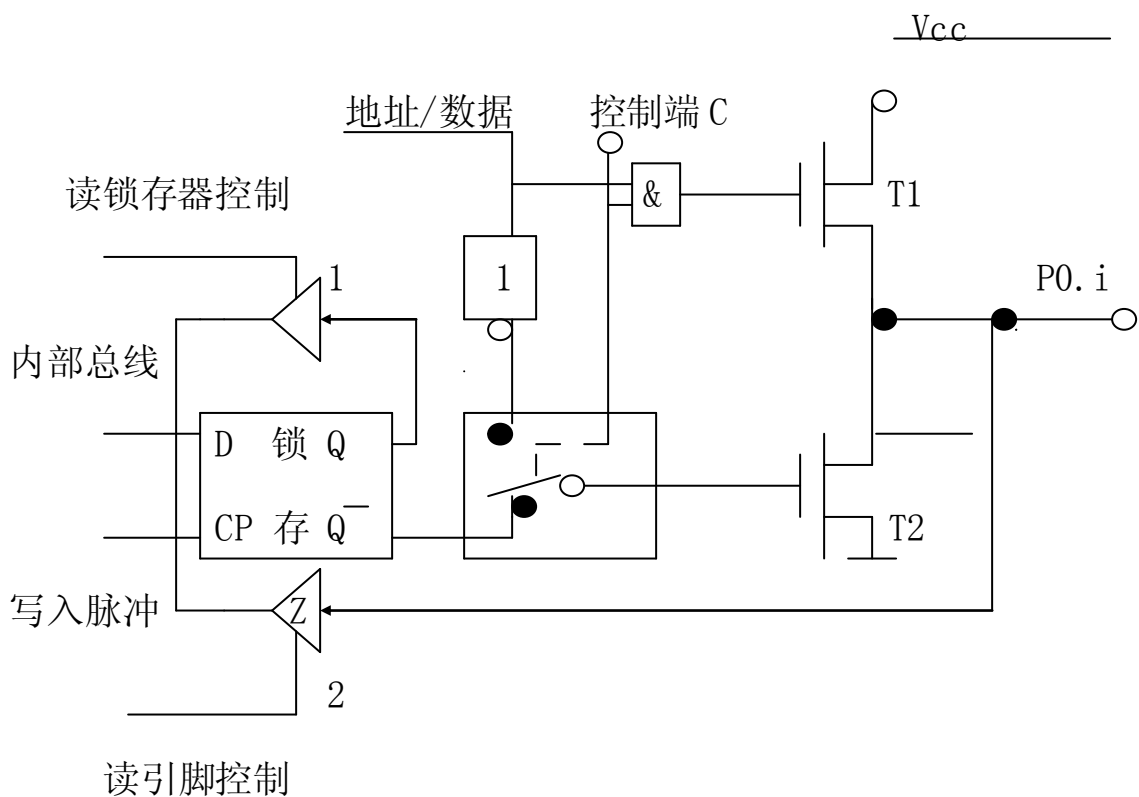
MCS-51 单片机内部属单总线结构，因此使系统在结构上增加了灵活性。通过总线，用户可根据应用需要进行多功能的系统扩展，构成用户的实际应用系统。MCS-51 系列中的 8031 单片机，因其内部在结构上无程序存储器，所以它的应用系统必定为一个扩展的系统。因此，MCS-51 的 4 个并行 I/O 接口中的 P0、P2、P3 口基本上都具备有这两项功能。

a、P0 口：P0 口的内部一位结构如图（三一3）所示。P0 口是一个多功能口除可以作为通用的输入/输出口外，还具备用于系统扩展的第二功能。在 MCS-51 的进行系统扩展时，它作为地址/数据总线口。通过外接地址锁存器，MCS-51 的内部单总线可从 P0 口被扩展成 8 位的数据总线和 16 位地址总线的低 8 位。在实际应用中，P0 口先送出外部存储器 16 位地址中的低 8 位至地址锁存器锁存，然后再由 P0 口进行 8 位数据的输入或输出。

b、P1 口：P1 口作为通用 I/O 接口，它的每一位都可以别编程为通用 I/O 接口线。

c、P2 口：P2 口也是一个多功能口，与 P0 口相似，它除可被用作 I/O 接口外，在进行系统扩展时，还可以输 16 位地址总线中的高 8 位，和 P0 口共同构成 16 位的地址总线。当然，在 P0 口和 P2 口用作地址/数据总线时，它们都不能再作为通用 I/O 接口。

d、P3 口：P3 口也是一个多功能口，除可以作为通用 I/O 接口外，还具有多种控制功能，为通用 I/O 接口时和 MCS-51 其他具有控制功能的输入/输出引线在一起，共同形成 MCS-51 的控制总线。P3 口在作为第二功能（控制功能）使用时，它的每一位功能定义如表（三一—2）所示



图（三一—3） P0 口内部一位结构图

一个信号引脚，既是第一功能又是第二功能，在使用时也不会引起混乱和造成错误，理由如下：

a、对于各种型号的芯片，其功能的第一功能信号是相同的，所不同的只在引脚的第二功能信号上。

b、对于 9、30 和 31 各引脚，由于第一功能信号与第二功能信号是单片机在不同工作方式下的信号，因此不会发生使用上的矛盾。

c、P3 口线的情况却有所不同，它的第二功能信号都是单片机的重要控制信号。因此在实际使用时，总是先按需要优先选用它的二功能，剩下不用的才作为口线使用。

表（三一2） P3 口线的第二功能

口线	第二功能	信号名称
P3.0	RXD	串行数据接收
P3.1	TXD	串行数据发送
P3.2	$\overline{\text{INT0}}$	外部中断 0 申请
P3.3	$\overline{\text{INT1}}$	外部中断 1 申请
P3.4	T0	定时器/计数器 0 计数输入
P3.5	T1	定时器/计数器 1 计数输入
P3.6	$\overline{\text{WR}}$	外部 RAM 写选通
P3.7	$\overline{\text{RD}}$	外部 RAM 读选通

d、引脚表现出的是单片机的外特性或硬件特性，在硬件方面用

户只能使用引脚，即通过引脚组建系统。因此熟悉引脚是我们设计数字显示温度计的重要内容。

#### 4、MCS-51 单片机的复位方式和复位电路

##### (1) 复位操作

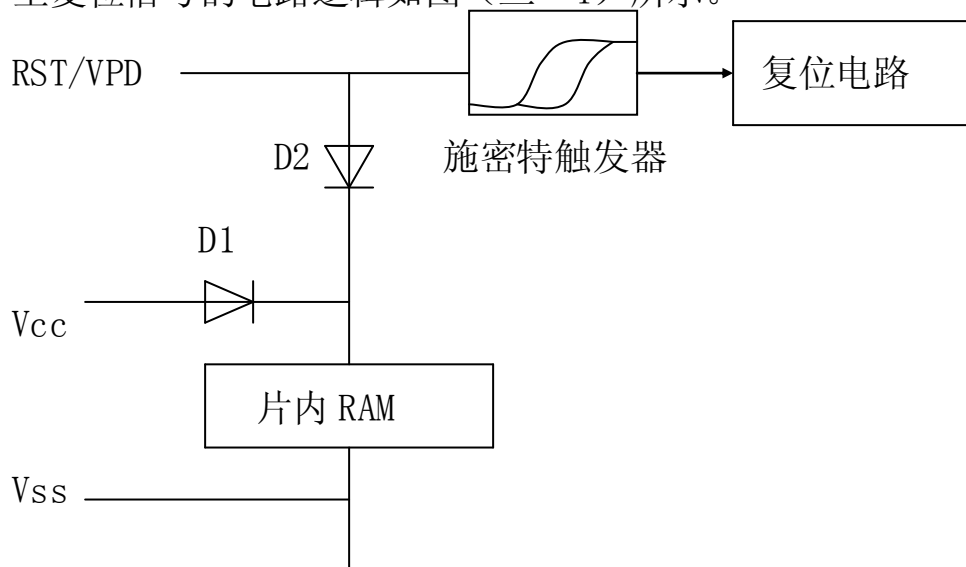
复位是单片机的初始化操作，其主要功能是把 PC 初始化为 0000H，使单片机从 0000H 单元开始执行程序。除了进入系统的正常初始化之外，当由于程序运行出错或者操作错误使系统处于死锁状态时，为摆脱困境，也需按复位键以重新启动。

(1) 复位操作还对单片机的个别引脚信号有影响，例如把 ALE 和 PSEN 非信号变为无效状态，即  $ALE=0$ ， $\overline{PSEN}=1$ 。

##### (2) 复位信号及其产生

RST 引脚是复位信号的输入端，复位信号是高电平有效，其有效时间持续 24 个振荡脉冲周期（即 2 个机器周期）以上，若使用频率为 6MHz 的晶振，则复位信号持续时间应超过 4us 才能完成复位操作。

产生复位信号的电路逻辑如图（三—4）所示。



图（三一4） 复位电路逻辑图

整个复位电路包括芯片内外两部分。外部电路产生的复位信号（RST）送施密特触发器，再由片内复位电路在每个机器周期的 S5P2 时刻对施密特触发器的输出进行采样。然后才得到内部复位操作所需要的信号。

### （3）复位方式

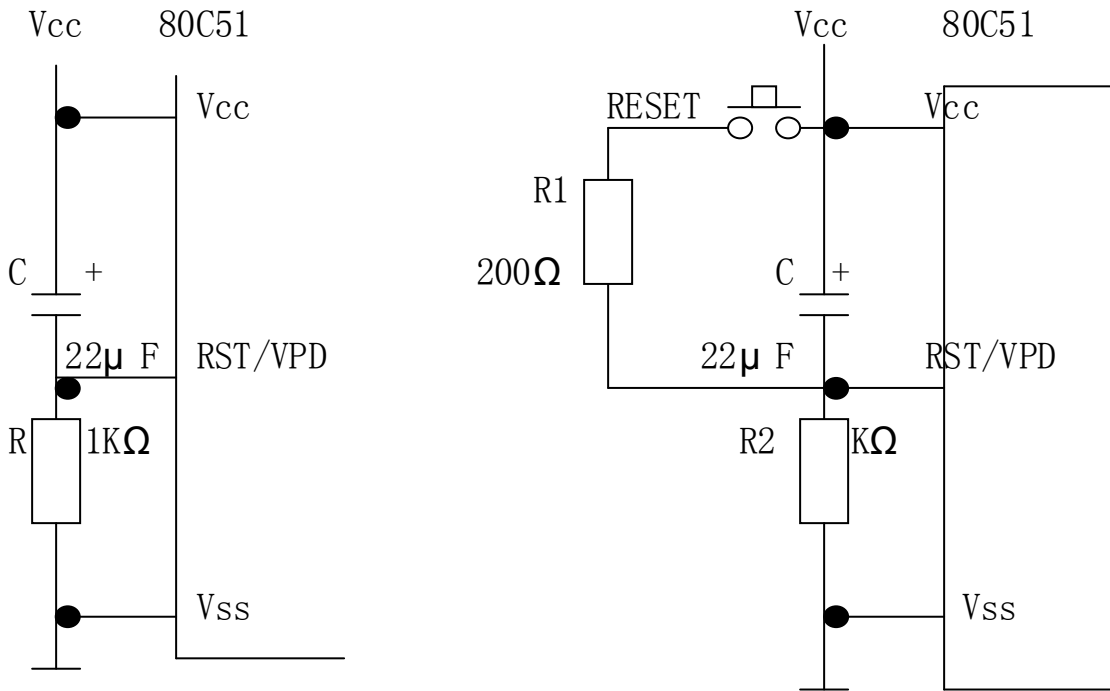
复位操作有上电自动复位和按键手动复位两种方式。上电自动复位是通过外部复位电路的电容充电来实现的，在单片机控制的数字显示温度计电路设计中的单片机复位就是这种方式，其中电容 C 为 22 $\mu$ F，电阻 R 为 1k 欧姆，其电路图如图（三一5a）所示。

这样，只要电源  $V_{cc}$  (+5V) 的上升时间不超过 1ms，就可以实现自动上电复位，即接通电源就完成了系统的复位初始化。

按键手动复位有电平方式和脉冲方式两种。其中按键电平复位是通过使复位端经电阻与  $V_{cc}$  电源接通而实现的，其电路图如图（三一5b）所示。而按键复位脉冲复位则是利用 RC 微分电路产生的正脉冲来实现的。

上述电路图中的电阻电容参数适宜于 6MHz 晶振，能保证复位信号高电平持续时间大于 2 个机器周期。



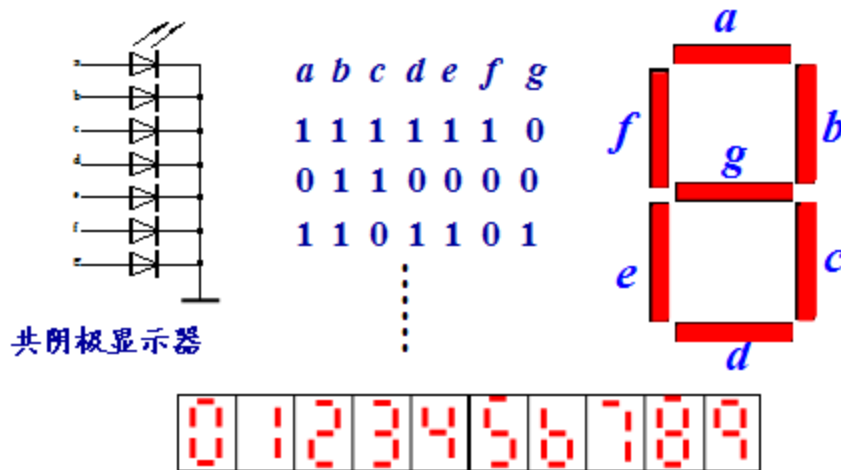


(a) 上电复位

(b) 按键电平复位

### 5 显示温度值的 LED 显示器接口介绍

LED 显示器是单片机应用系统中常用的廉价输出设备。它是由若干个发光二极管组成的，当发光二极管导通时，相应一个笔画划发光，控制某段发光二极管导通，就能显示出某个数码或字符，常用八段 LED 显示器有两种结构，如图（四—1）所示。



图（四—1） 八段 LED 显示器的结构

在静态显示系统中,每位显示器都应有各自的锁存器、译码器(若采用软件译码,译码器可省去)与驱动器,用以锁存各自待显示数字的BCD码或字段码。因此,静态显示系统在每一次显示输出后能够保持显示不变,仅在待显示数字需要改变时,才更新其数字显示锁存器中的内容。这种显示占用CPU的时间少,显示稳定可靠。缺点是,当显示的位数较多时,占用的I/O口较多。

在动态显示的系统中,CPU需定时地对每位LED显示器进行扫描,每位LED显示器分时轮流工作,每次只能使一位LED显示,但由于人的视觉暂留现象,仍感觉所有的LED显示器都在同时显示。这种显示的优点是使用硬件少,占用I/O口少。缺点是占用CPU时间长,只要不执行显示程序,就立刻停止显示。但随着大规模集成电路的发展,目前已有能自动对显示器进行扫描的专用显示芯片,使电路既简单又占用CPU时间。在我们所设计的温度计中数码管显示就是利用的动态显示。

## 四 温度传感器 DS1820 介绍

### (一) DS1820 芯片简介:

在传统的模拟信号远距离温度测量系统中,需要很好的解决引线误差补偿问题、多点测量切换误差问题和放大电路零点漂移误差问题等技术问题,才能够达到较高的测量精度。我们在为某水电站开发水轮发电机组轴瓦温度实时监测系统时,为了克服上面提到的三个问题,采用了新型数字温度传感器 DS1820,在对其测温原理进

行详细分析的基础上,提出了提高DS1820测量精度的方法,使DS1820的测量精度由 $0.5^{\circ}\text{C}$ 提高到 $0.1^{\circ}\text{C}$ 以上,取得了良好的测温效果。

## 1. DS1820 简介

DS1820 是美国 DALLAS 半导体公司生产的可组网数字式温度传感器,在其内部使用了在板(ON-BOARD)专利技术。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。与其它温度传感器相比,DS1820 具有以下特性:

- (1) 独特的单线接口方式,DS1820 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DS1820 的双向通讯。
- (2) DS1820 支持多点组网功能,多个DS1820 可以并联在唯一的三线上,实现多点测温。
- (3) DS1820 在使用中不需要任何外围元件。
- (4) 温范围 $-55^{\circ}\text{C}\sim+125^{\circ}\text{C}$ ,固有测温分辨率 $0.5^{\circ}\text{C}$ 。
- (5) 测量结果以9位数字量方式串行传送。

2. DS1820 内部结构框图如图 1 所示。

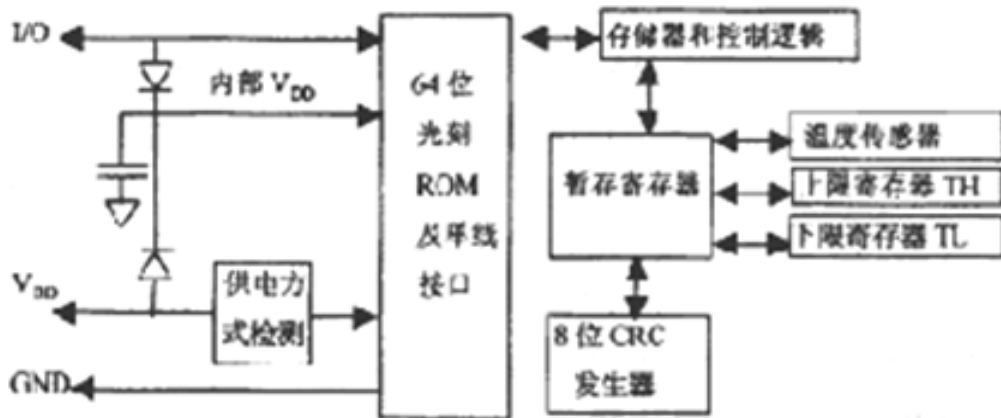


图 1 DS1820 内部结构框图

DS1820 测温原理如图 2 所示。图中低温度系数晶振的振荡频率受温度影响很小，用于产生固定频率的脉冲信号送给计数器 1。高温系数晶振随温度变化其振荡率明显改变，所产生的信号作为计数器 2 的脉冲输入。计数器 1 和温度寄存器被预置在  $-55^{\circ}\text{C}$  所对应的一个基数值。计数器 1 对低温度系数晶振产生的脉冲信号进行减法计数，当计数器 1 的预置值减到 0 时，温度寄存器的值将加 1，计数器 1 的预置将重新被装入，计数器 1 重新开始对低温度系数晶振产生的脉冲信号进行计数，如此循环直到计数器 2 计数到 0 时，停止温度寄存器值的累加，此时温度寄存器中的数值即为所测温度。图 2 中的斜率累加器用于补偿和修正测温过程中的非线性，其输出用于修

正计数器 1 的预置值。



图 2 DS1820 测温原理框图

在正常测温情况下，DS1820 的测温分辨率为  $0.5^{\circ}\text{C}$  以 9 位数据格式表示，其中最低有效位 (LSB) 由比较器进行  $0.25^{\circ}\text{C}$  比较，当计数器 1 中的余值转化成温度后低于  $0.25^{\circ}\text{C}$  时，清除温度寄存器的最低位 (LSB)，当计数器 1 中的余值转化成温度后高于  $0.25^{\circ}\text{C}$ ，置位温度寄存器的最低位 (LSB)，如  $-25.5^{\circ}\text{C}$  对应的 9 位数据格式如下：

### 3. 提高 DS1820 测温精度的途径

#### (1) DS1820 高精度测温的理论依据

DS1820 正常使用时的测温分辨率为  $0.5^{\circ}\text{C}$ ，这对于水轮发电机组轴瓦温度监测来讲略显不足，在对 DS1820 测温原理详细分析的基础上，我们采取直接读取 DS1820 内部暂存寄存器的方法，将 DS1820 的测温分辨率提高到  $0.1^{\circ}\text{C} \sim 0.01^{\circ}\text{C}$ 。

DS1820 内部暂存寄存器的分布如表 1 所示，其中第 7 字节存放的是当温度寄存器 停止增值时计数器 1 的计数剩余值，第 8 字节存放的是每度所对应的计数值，这样，我们就可以通过下面的方法获得高分辨率的温度测量结果。首先用 DS1820 提供的读暂存寄存器指令 (BEH) 读出以 0.5℃ 为分辨率的温度测量结果，然后切去测量结果中的最低有效位 (LSB)，得到所测实际温度整数部分 T 整数，然后再用 BEH 指令读取计数器 1 的计数剩余值 M 剩余和每度计数值 M 每度，考虑到 DS1820 测量温度的整数部分以 0.25℃、0.75℃ 为进位界限的关系，实际温度 T 实际可用下式计算得到：

$$T_{\text{实际}} = (T_{\text{整数}} - 0.25^{\circ}\text{C}) + (M_{\text{每度}} - M_{\text{剩余}}) / M_{\text{每度}}$$

表 1 DS1820 暂存寄存器分布

寄存器内容	字节地址
温度最低数字位	0
温度最高数字位	1
高温限值	2
低温限值	3
保留	4
保留	5
计数剩余值	6
每度计数值	7
CRC 校验	8

## (2) 测量数据比较

表 2 为采用直接读取测温结果方法和采用计算方法得到的测温数据比较，通过比较可以看出，计算方法在 DS1820 测温中不仅是可行的，也可以大大的提高 DS1820 的测温分辨率。

#### 4. DS1820 使用中注意事项

DS1820 虽然具有测温系统简单、测温精度高、连接方便、占用口线少等优点，但在实际应用中也应注意以下几方面的问题：

(1) 较小的硬件开销需要相对复杂的软件进行补偿，由于 DS1820 与微处理器间采用串行数据传送，因此，在对 DS1820 进行读写编程时，必须严格的保证读写时序，否则将无法读取测温结果。在使用 PL/M、C 等高级语言进行系统程序设计时，对 DS1820 操作部分最好采用汇编语言实现。

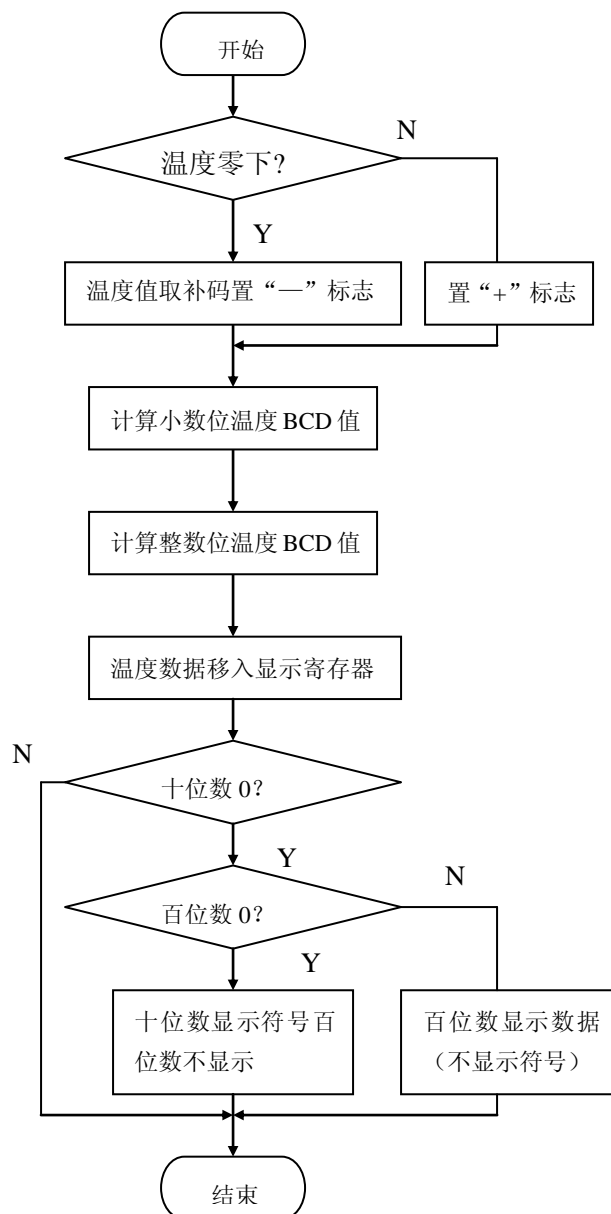
(2) 在 DS1820 的有关资料中均未提及单总线上所挂 DS1820 数量问题，容易使人误认为可以挂任意多个 DS1820，在实际应用中并非如此。当单总线上所挂 DS1820 超过 8 个时，就需要解决微处理器的总线驱动问题，这一点在进行多点测温系统设计时要加以注意。

(3) 连接 DS1820 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，读取的测温数据将发生错误。当将总线电缆改为双绞线带屏蔽电缆时，正常通讯距离可达 150m，当采用每米绞合次数更多的双绞线带屏蔽电缆时，正常通讯距离进一步加长。这种情况主要是由总线分布电容使信号波形产生畸变造成的。因此，在用 DS1820 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。

(4) 在 DS1820 测温程序设计中，向 DS1820 发出温度转换命令后，程序总要等待 DS1820 的返回信号，一旦某个 DS1820 接触不好或断线，当程序读该 DS1820 时，将没有返回信号，程序进入死循环。这一点在进行 DS1820 硬件连接和软件设计时也要给予一定的重视。

## 五 软件编程

### 1 温度控制流程图





## 2 电路程序

```

;6M 晶体 89c51
temphead    equ    40h
display_data1_reg data  30h
display_data2_reg data  31h
display_data3_reg data  32h
display_long_reg  data  33h
led1_bit     bit    p3.0
led2_bit     bit    p3.1
led3_bit     bit    p3.2
timelsok     bit    01H
TEMPONEOK    BIT    02H
wiredogcs           bit           p3.7
;TIAOSHI_LONG     DATA  21H
;TIAOSHI_DATA     DATA  22H
templ          data  34h
tempb          data  35h
tempc          data  36h
templc         data  37h
tempdin        bit    p2.0
    ORG    0000H
        LJMP  START
    ORG    000BH
        AJMP  LED
    ORG    0030H
start:    MOV    sp, #60H
        mov    tmod, #21H
        mov    pcon, #00H
        mov    scon, #01010000b
        mov    th1, #243
        mov    tl1, #243
        MOV    TH0, #0FCH
        MOV    TL0, #18H
        SETB  TR0
        setb  tr1

```

```

        SETB  ET0
        mov  display_long_reg, #00h
        SETB  EA
MAIN:    CLR   TEMPONEOK
MAIN1:   LCALL DLY_1S
        JNB  TEMPONEOK, MAIN2
        clr  ea
        LCALL READTEMP1
        setb ea
        LCALL CONVTEMP
MAIN2:   clr  ea
        LCALL READTEMP
        setb ea
        setb TEMPONEOK
        AJMP MAIN1
;*****
;      DS1820 复位
;*****
INITDS1820:  SETB  TEMPDIN
            NOP
            NOP
            CLR   TEMPDIN
            MOV   R6, #0A0H
            DJNZ  R6, $
            MOV   R6, #0A0H
            DJNZ  R6, $
            SETB  TEMPDIN
            MOV   R6, #32H
            DJNZ  R6, $
            MOV   R6, #3CH
LOOP1820:  MOV   C, TEMPDIN
            JC   INITDS1820OUT
            DJNZ  R6, LOOP1820
            MOV   R6, #64H
            DJNZ  R6, $
            SJMP  INITDS1820
            RET

```

```

INITDS1820OUT:   SETB  TEMPDI
                 RET
;*****
;           读 DS1820 的数据
;入口条件: 数据存放在 ACC 中
;*****
READDS1820:     MOV   R7, #08H
                 SETB  TEMPDI
                 NOP
                 NOP
READDS1820LOOP: CLR   TEMPDI
                 NOP
                 NOP
                 NOP
                 SETB  TEMPDI
                 MOV   R6, #07H
                 DJNZ  R6, $
                 MOV   C, TEMPDI
                 MOV   R6, #3CH
                 DJNZ  R6, $
                 RRC   A
                 SETB  TEMPDI
                 DJNZ  R7, READDS1820LOOP
                 MOV   R6, #3CH
                 DJNZ  R6, $
                 RET
*****
;           写 DS1820 的数据
;出口条件: 数据存放在 ACC 中
;*****
WRITEDS1820:     MOV   R7, #08H
                 SETB  TEMPDI
                 NOP
                 NOP
WRITEDS1820LOP: CLR   TEMPDI
                 MOV   R6, #07H
                 DJNZ  R6, $

```

```

RRC    A
MOV    TEMPDIN, C
MOV    R6, #34H
DJNZ   R6, $
SETB   TEMPDIN
DJNZ   R7, WRITEDS1820LOP
RET

;*****
;      启动温度转换
;*****
READTEMP:    LCALL INITDS1820
              MOV    A, #0CCH
              LCALL WRITEDS1820
              MOV    R6, #34H
              DJNZ   R6, $
              MOV    A, #44H
              LCALL WRITEDS1820
              MOV    R6, #34H
              DJNZ   R6, $
              RET

;*****
;      读取转换后的温度值
;*****
READTEMP1:    LCALL INITDS1820
              MOV    A, #0CCH
              LCALL WRITEDS1820
              MOV    R6, #34H
              DJNZ   R6, $
              MOV    A, #0BEH
              LCALL WRITEDS1820
              MOV    R6, #34H
              DJNZ   R6, $
              MOV    R5, #09H
              MOV    R0, #TEMPHEAD
              MOV    B, #00H
READTEMP2:    LCALL READDS1820
              MOV    @R0, A

```

```

                INC    R0
READTEMP21:    LCALL  CRC8CAL
                DJNZ  R5, READTEMP2
                MOV   A, B
                JNZ  READTEMPOUT
                MOV   R1, #TEMPHEAD
                MOV   A, @R1
                MOV   TEMPL, A
                INC   R1
                MOV   A, @R1
                MOV   TEMPH, A
READTEMPOUT:   RET
;*****
;           处理温度 BCD 码程序
;*****
CONVTEMP:     MOV   A, TEMPH
                ANL   A, #80H
                JZ   TEMPC1      ;判断是否为负数
                CLR   C
                MOV   A, TEMPL
                CPL   A
                ADD  A, #01H
                MOV   TEMPL, A
                MOV   A, TEMPH
                CPL   A
                ADDC A, #00H
                MOV   TEMPH, A
                MOV   TEMPHC, #0BH
                SJMP  TEMPC11
TEMPC1:       MOV   TEMPHC, #0AH
TEMPC11:     MOV   A, TEMPHC
                SWAP  A
                MOV   TEMPHC, A
                MOV   A, TEMPL
                ANL   A, #0FH
                MOV   DPTR, #TEMPDOTTAB
                MOVC  A, @A+DPTR

```

```

MOV    TEMPLC, A
MOV    A, TEMPL
ANL    A, #0FOH
SWAP   A
MOV    TEMPL, A
MOV    A, TEMPH
ANL    A, #0FH
SWAP   A
ORL    A, TEMPL
LCALL  HEX2BCD1
MOV    TEMPL, A
ANL    A, #0FOH
SWAP   A
ORL    A, TEMPHC
MOV    TEMPHC, A
MOV    A, TEMPL
ANL    A, #0FH
SWAP   A
ORL    A, TEMPLC
MOV    TEMPLC, A
MOV    A, R7
JZ     TEMPC12
ANL    A, #0FH
SWAP   A
MOV    R7, A
MOV    A, TEMPHC
ANL    A, #0FH
ORL    A, R7
MOV    TEMPHC, A
TEMPC12:    RET
TEMPDOTTAB:    DB 00H, 01H, 01H, 02H, 03H, 03H, 04H, 04H, 05H, 06H
                DB 06H, 07H, 08H, 08H, 09H, 09H
;*****
;    CRC 校验程序
;*****
CRC8CAL:    PUSH  ACC
            MOV   R7, #08H

```

```

CRC8LOOP1:    XRL   A, B
              RRC   A
              MOV   A, B
              JNC   CRC8LOOP2
              XRL   A, #18H
CRC8LOOP2:    RRC   A
              MOV   B, A
              POP   ACC
              RR   A
              PUSH  ACC
              DJNZ  R7, CRC8LOOP1
              POP   ACC
              RET

;*****
;    单字节 16 进制转换 BCD
;入口条件: 数据存放在 ACC 中
;出口条件: 数据存放在 ACC 中
;*****
HEX2BCD1:    MOV   B, #64H
              DIV   AB
              MOV   R7, A
              MOV   A, #0AH
              XCH   A, B
              DIV   AB
              SWAP  A
              ORL   A, B
              RET

;*****
;    1S 延时程序
;*****
DLY_1S:      PUSH  PSW
              SETB  RS0
              MOV   R7, #10
DLY_1S_1:    MOV   R6, #200
              clr   wiredogcs ;喂狗
              setb  wiredogcs
DLY_1S_2:    MOV   R5, #250

```

```

        DJNZ  R5, $
        DJNZ  R6, DLY_1S_2
        DJNZ  R7, DLY_1S_1
        POP   PSW
        RET

;*****
;      数码管显示程序
;*****

;*****
;      LED
;入口条件：输入数据放入 ACC
;输出：转换后的数据保存显示器缓冲区中中
;影响数据：ACC
;*****
LED:      push  acc
          push  dph
          push  dpl
          mov   th0, #0fch
          mov   t10, #66h
          mov   dptr, #led_tab_1
          mov   a, display_long_reg
          inc   display_long_reg
          rl   a
          jmp   @a+dptr
led_tab_1:  ajmp  led1
          ajmp  led2
          ajmp  led3
led1:      setb  led1_bit
          setb  led2_bit
          setb  led3_bit
          mov   a, TEMPLC
          anl   a, #0fh
          mov   dptr, #led_tab_2
          movc  a, @a+dptr
          mov   p1, a
          clr   led3_bit

```



```

        ajmp  led_exit
led2:   setb  led1_bit
        setb  led2_bit
        setb  led3_bit
        mov   a, TEMPLC
        anl   a, #0f0h
        swap a
        mov   dptr, #led_tab_2
        movc  a, @a+dptr
        clr   acc.7
        mov   p1, a
        clr   led2_bit
        ajmp  led_exit
led3:   setb  led1_bit
        setb  led2_bit
        setb  led3_bit
        mov   a, TEMPHC
        anl   a, #0fh
        mov   dptr, #led_tab_2
        movc  a, @a+dptr
        mov   p1, a
        clr   led1_bit
        mov   display_long_reg, #0
        ajmp  led_exit
led_tab_2: DB      0C0H, 0F9H, 0A4H, 0B0H, 99H
           DB      92H, 82H, 0F8H, 80H, 90H
           DB      88H, 83H, 0C6H, 0A1H, 86H, 8EH, 0ffH
led_exit: pop   dpl
           pop   dph
           pop   acc
           reti
;*****
;      485 发送程序
;*****
        END

```

## 致 谢

在毕业论文接近末尾之时，我要衷心地感谢我们罗木贵老师，在我整个毕业设计过程中，罗木贵老师给了我很大的帮助和细心的指导。在一个多月的毕业设计过程中，当我遇到了困难和问题时，当我们需要他的时候，罗木贵老师总是第一时间出现在我们面前，他让我们学会了以前在课堂上没有的东西。

另外，我还要特别感谢我所有的搭档，是他们给了我巨大的勇气和战胜困难的信心，在毕业设计中我们合作的很愉快，当我们遇到困难时我们一起去探讨和研究，一起去战胜它，大家也一起分享排除问题和困难后的喜悦。

同时大家也发扬我们慷慨激扬精神：特别能吃苦，特别能攻关，特别能战斗，特别能奉献。

在此我表示真诚的感谢！

## 心得体会

经过这次毕业设计，使我觉得不论从理论知识还是从实际操纵中都学到了不少知识，我想归纳起来，主要有以下四个方面：

1、经过这次毕业设计，它让我接触更多平时没有接触过的科学仪器设备、元器件以及获得相关的仪器调试经验，同时我也发现自己在这方面很多不足之处。体会到理论知识对实践有很大的指导作用，她让我知道，只有在正确的理论指引下，才能设计出合乎实际需要的硬件电路。

2、学会了高效率的查阅资料、运用工具书、利用网络查找资料。我发现，在我们所使用的书籍上有一些知识在实际应用中其实并不是十分理想，各种参数都需要自己去调整。偶而还会遇到错误的资料现象，这就要求我们应更加注重实践环节。

3、在毕业设计中，我们应当注意重点与细节的关系。

4、失败不可怕，只要不趴下，昂首向前走，希望总会有。

5、同组同学相互包容，彼此合作，取长补短，才能铸就最后的成功。

可以这样说毕业设计是对大学三年所学知识的一次运用和检阅，同时对自学能力提出很高的要求，所以平时的学习离开思考，就是严重的错误，我们学习不应该有偏科现象，各方面的知识都应该要接触，这样做才能为毕业设计打下基石。

## 参考文献

- [1] 李广弟等编 单片机基础 北京航空航天大学出版社 2001
- [2] 吴金戌等编 8051 单片机实践与应用  
清华大学出版社 2001
- [3] 王洪庆编 微型计算机控制技术  
机械工业出版社 2006
- [4] 付家才编 单片机控制工程实践技术  
化学工业出版社 2004
- [5] 鲍宏亚等编 MCS-51 系列单片机应用系统设计及实用技术  
中国宇航出版社 2005