

---

# 硅湖职业技术学院毕业论文（设计）

题目 浅谈智能温度报警器的设计

年级 091

专业 机电一体化

姓名 薛龙

学号 090000608

指导老师 潘世丽

2012年5月10日

# 浅谈智能温度报警器的设计

薛龙

**【摘要】**本文主要介绍了一个基于AT89S51单片机的测温系统,详细描述了利用数字温度传感器DS18B20开发测温系统的过程,重点对传感器在单片机下的硬件连接,汇编语言编程以及各系统模块流程进行了详尽分析,对各部分的电路也一一进行了介绍,对单片机学习有很强的实用价值和意义。

**【关键词】**单片机 温度传感器 汇编语言

## 1 引言

随着微电子技术和微型计算机的迅猛发展,微机测量和控制技术以其逻辑简单、控制灵活、使用方便及性能价格比高的优点得到了迅猛发展和广泛应用。它不仅在航空、航天、铁路交通、冶金、电力、电讯、石油化、制造业等领域获得了广泛应用,而且其技术在日常生活中诸如微波炉、电冰箱、电视机、电动玩具、智能空调机等高科技产品也具有广阔的应用前景。尤其是许多智能仪表和测控系统中电脑控制技术的引入,使得传统仪器、仪表设备发生了根本变化,为工业生产的自动化、智能化奠定了坚实的技术基础。

在工业生产、石油化工、电力、国防、科研领域和日常生活中,温度是极为普遍又极为重要的热工参数之一,温度是表征物体冷热程度的物理量,它是一个很重要的物理参数,也是一个非电物量。自然界中任何物理、化学过程都紧密地与温度相联系。温度的测量与控制直接和产品质量、提高生产率、节约能源、安全生产等重要经济技术指标相联系,因此温度测量是一个具有重要意义的技术领域。

温度传感器是在温度测量系统使用数量大、应用最广泛的一种传感器。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,该设计控制器使用单片机AT89S51,测温传感器使用DS18B20,用共阳极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。本文主要讨论单总线数字集成温度传感器DS18B20的特点、工作原理和使用方法,以及用单片机AT89S51对DS18B20的编程实现温度测量,有很强的学习价值和意义。

## 2 方案选择

### 2.1 方案一

由于本设计是测温电路，可以使用热敏电阻之类的器件利用其感温效应，再将随被测温度变化的电压或电流采集过来，进行 A/D 转换后，就可以用单片机进行数据的处理，在显示电路上，就可以将被测温度显示出来，这种设计需要用到 A/D 转换电路，感温电路比较麻烦。

### 2.2 方案二

基于上述原因，进而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。从以上两种方案，很容易看出，采用方案二，电路比较简单，软件设计也比较简单，故采用了方案二。

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 LED 数码管以串行口传送数据实现温度显示。

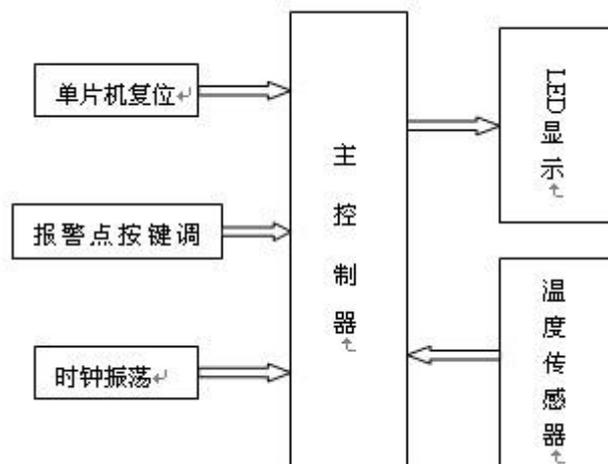


图 1 总体设计方框图

## 3 芯片介绍

### 3.1 主控制器

单片机 AT89S51 具有低电压供电和体积小等特点，四个端口只需要两个口就能满足电路系统的设计需要，很适合便携带手持式产品的设计使用系统可用二节电池供电。AT89S51 是一个低功耗，高性能 CMOS 8 位单片机，片内含

8k Bytes ISP 的可反复擦写 1000 次的 Flash 只读程序存储器，器件采用 ATMEL 公司的高密度、非易失性存储技术制造，兼容标准 MCS-51 指令系统及 80C51 引脚结构，芯片内集成了通用 8 位中央处理器和 ISP Flash 存储单元，功能强大的微型计算机的 AT89S51 可为许多嵌入式控制应用系统提供高性价比的解决方案。AT89S51 具有如下特点：40 个引脚，4k Bytes Flash 片内程序存储器，128 bytes 的随机存取数据存储器（RAM），32 个外部双向输入/输出（I/O）口，5 个中断优先级 2 层中断嵌套中断，2 个 16 位可编程定时计数器，2 个全双工串行通信口。

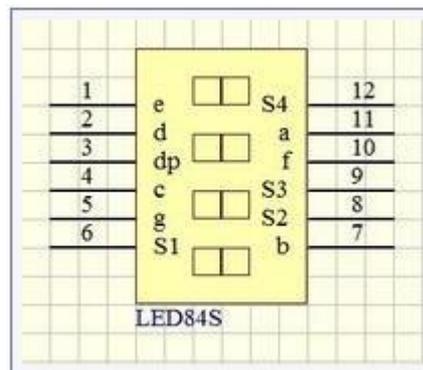


图 2 LED84S 引脚图

### 3.2 数码管 LED84S

数码管 LED84S 的引脚图如图 2 所示，其中  $S_{1,2,3,4}$  就是 4 个 LED 的公共端，如果是共阳的话你想要第一个“日”亮，就给 S1 接电源的正极，a. b. c. d. e. f. g 就是这个“日”的各个笔划，叫做段选， $S_{1,2,3,4}$  分别是位选。

### 3.3 温度传感器 DS18B20

DS18B20 温度传感器是美国 DALLAS 半导体公司最新推出的一种改进型智能温度传感器，与传统的热敏电阻等测温元件相比，它能直接读出被测温度，并且可根据实际要求通过简单的编程实现 9 ~ 12 位的数字值读数方式。DS18B20 的性能特点如下：

- 具有三引脚 PR. 35 封装形式。
- 具有独特的单总线接口方式，CPU 只需一根端口线就能与 DS18B20 通信。
- 温度测量范围为. 55°C---125. C。测温分辨率可达 0. 0625°C。
- 被测温度用可编程为 9 位~12 位 A / D 转换精度的数字量串行输出。
- 其工作电源既可在远端引入、也可采用寄生电源方式产生，电压范围为 3. 0V-5. 5V。
- 可以将多个 DS18B20 温度传感器挂接在一根总线上，即允许一条信号线上接数十乃至上百个数字式传感器，每个都有一个在 ROM 中的 64 位自己独特的芯片序列号，可实现多点温度的检测。

- 有负电压特性，电源极性接反时，温度计不会因发热而烧毁，只是不能正常工作。

以上特点使DS18B20非常适用于远距离多点温度检测系统。由它组成的温度测控系统非常方便，克服了模拟式传感器与微机接口时需要的A / D转换器及其它复杂外围电路的缺点，占用微处理器的端口较少，可节省大量的引线和逻辑电路，从而使测温装置与各传感器的接口变得十分简单，而且成本低、体积小、可靠性高。

### 3.4 数字式温度传感器DS18B20的结构

(1) 温度传感器DS18B20引脚与封装形式：

温度传感器DS18B20有两种封装方式：分别是TO. 92封装和8脚SOIC封装，如图3所示，图的左边是DS18820的TO. 92封装，图的右边是DS18B20的8脚SOIC封装，本课题采用的是第一种DS18B20 TO. 92封装形式。

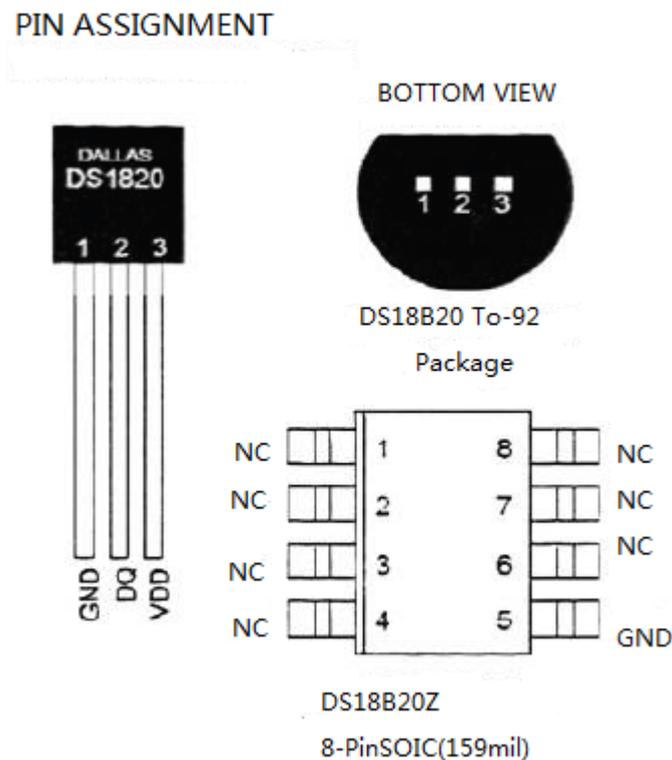


图 3 DS18B20 引脚排列

(2) 数字是温度转感器DS18B20的内部结构

DS18B20内部结构主要由四部分组成：64位光刻ROM，温度传感器，非挥发的温度报警触发器TH和TL，商速暂存器。64位光刻ROM是出厂前被光刻好的，它可以看作是該DS18B20的地址序列号。

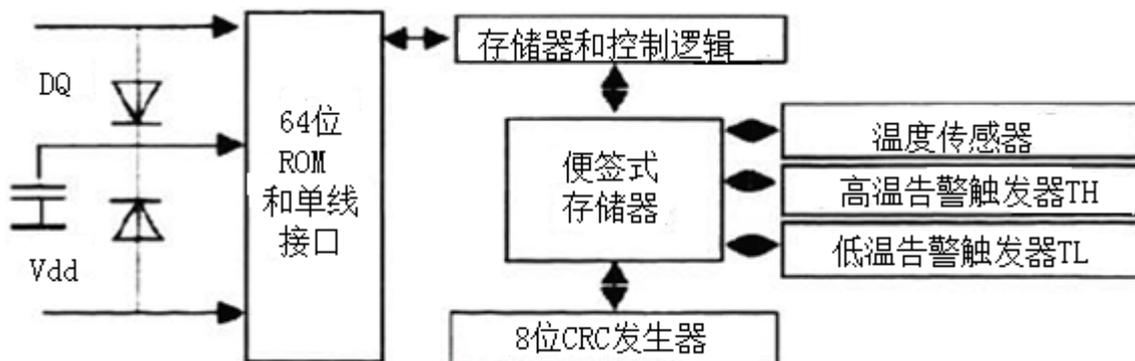


图 4 DS18B20 内部结构图

如图4所示. DQ为数字信号输入 / 输出端, GND为电源地; VDD为外接供电电源输入端。

ROM的作用是使每个DS18B20都并不相同, 这样就可以实现在总线挂多个DS18B20的目的。

非易失性温度报警触发器TH和TL, 可通过软件写入用户报警上下限数据。可通过软件写入用户报警上下限的数据。

高速暂存RAM的结构为9字节的存储器。前2字节包含测得的温度信息。第3、4字节是TH和TL的拷贝, 是易失的, 每次上电复位时被刷新。第5字节为配置寄存器, 其内容用于确定温度值的数字分辨率。DS18B20工作时按此寄存器中的分辨率转换为相应精度的数值。

如表5所示, 低5位一直为1; TM是测试位, 用于设置DS18B20在工作模式还是在测试模式, 在DS18B20出厂时, 该位被设置为0, 用户不要去改动; R1、R2决定温度转换的精度位数。

表 5 DS18B20 配置寄存器位定义

TM	R1	R2	1	1	1	1	1
----	----	----	---	---	---	---	---

以12位转化为例说明温度高低字节存放形式及计算: 12位转化后得到的12位数据, 存储在DS18B20的两个高低两个8位的RAM中, 二进制中的前面5位是符号位。如果测得的温度大于0, S这5位为0, 测得的温度值为正值, 可以直接将二进制转换为十进制; 如果温度小于0, S这5位为1, 测到的数值需要先将补码变成原码, 再计算十进制值, 才能得到实际温度。表6是反映了温度数据存放的格式。

表 6 温度数据格式

高8位	S	S	S	S	S	$2^6$	$2^5$	$2^1$
低8位	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$

这里举一例说明：比如测得实际的温度为30℃，那么在DS18B20表示的值为(0000 0001 1110 0000)<sub>2</sub>，它是由两个字节的二进制组成，高字节中的前四位为正负表示温度，低字节的后四位为温度的小数位，中间八位凑在一起就是用二进制表示实际测得温度的数值，即(0001 1110)<sub>2</sub>=30度。

### 3.5 移位寄存器 74LS164

74LS164 为 8 位移位寄存器,其内部结构如图 7 所示。当清除端（CLEAR）为低电平时，输出端（QA—QH）均为低电平。串行数据输入端（A, B）可控制数据。当 A、B 任意一个为低电平，则禁止新数据输入，在时钟端（CLOCK）脉冲上升沿作用下 Q0 为低电平。当 A、B 有一个为高电平，则另一个就允许输入数据，并在 CLOCK 上升沿作用下决定 Q0 的状态。

74LS164 电气参数:电源电压 7V; 输入电压 5.5V; 工作环境温度 0~70℃; 储存温度-65℃~150℃。其时序图如图 8 所示。

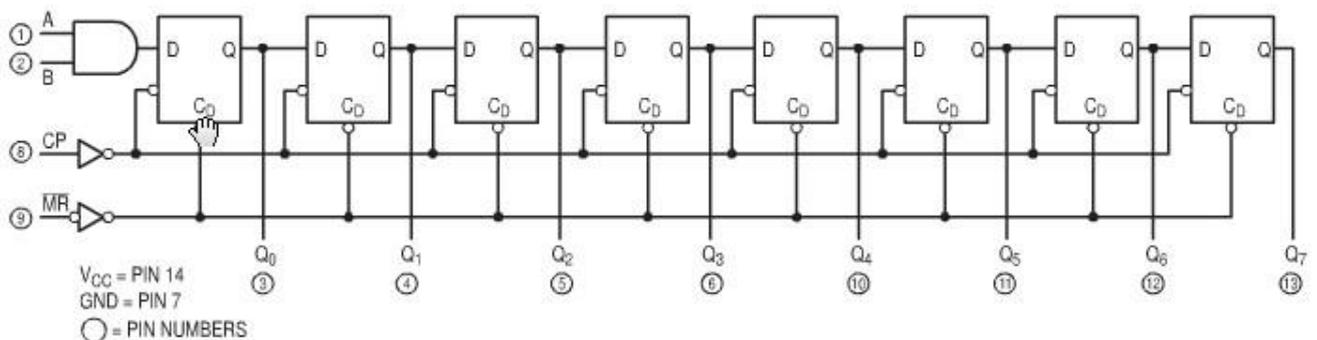


图 7 74LS164 内部结构图

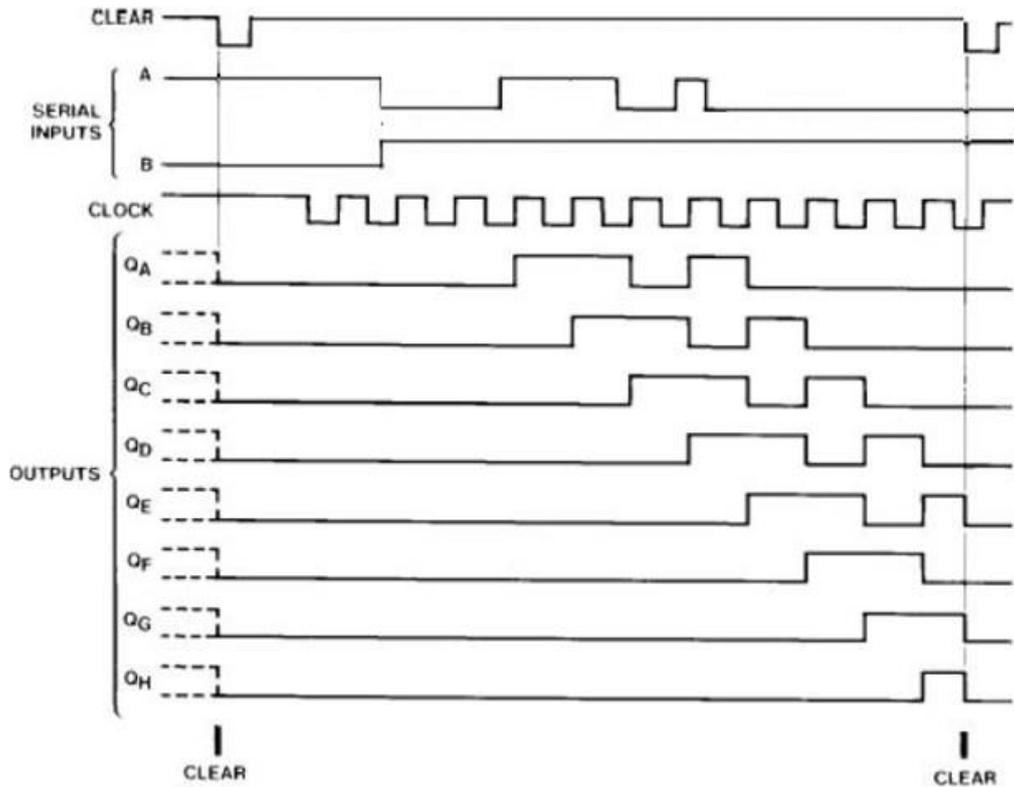


图 8 时序图

## 4 系统电路

系统整体硬件电路包括，传感器数据采集电路，温度显示电路，上下限报警调整电路，单片机主板电路等。

### 4.1 DS18B20 与单片机的接口电路

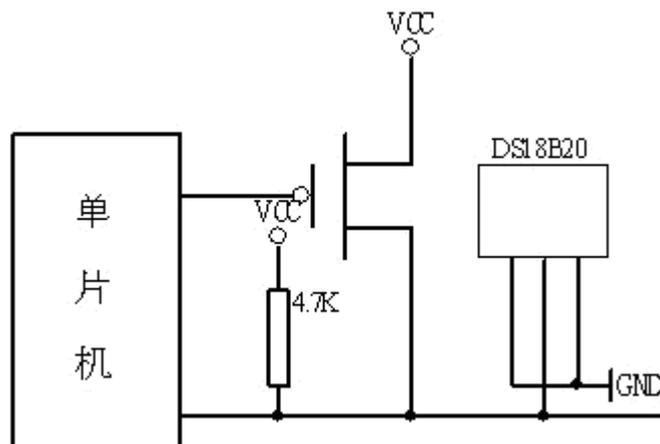


图 9 DS18B20 与单片机的接口电路

## 4.2 主板电路

图 10 中有三个独立式按键可以分别调整温度计的上下限报警设置，图中蜂鸣器可以在被测温度不在上下限范围内时，发出报警鸣叫声音，同时 LED 数码管将没有被测温度值显示，这时可以调整报警上下限，从而测出被测的温度值。图 10 中的按键复位电路是上电复位加手动复位，使用比较方便，在程序跑飞时可以手动复位，这样就不用重起单片机电源，就可以实现复位。

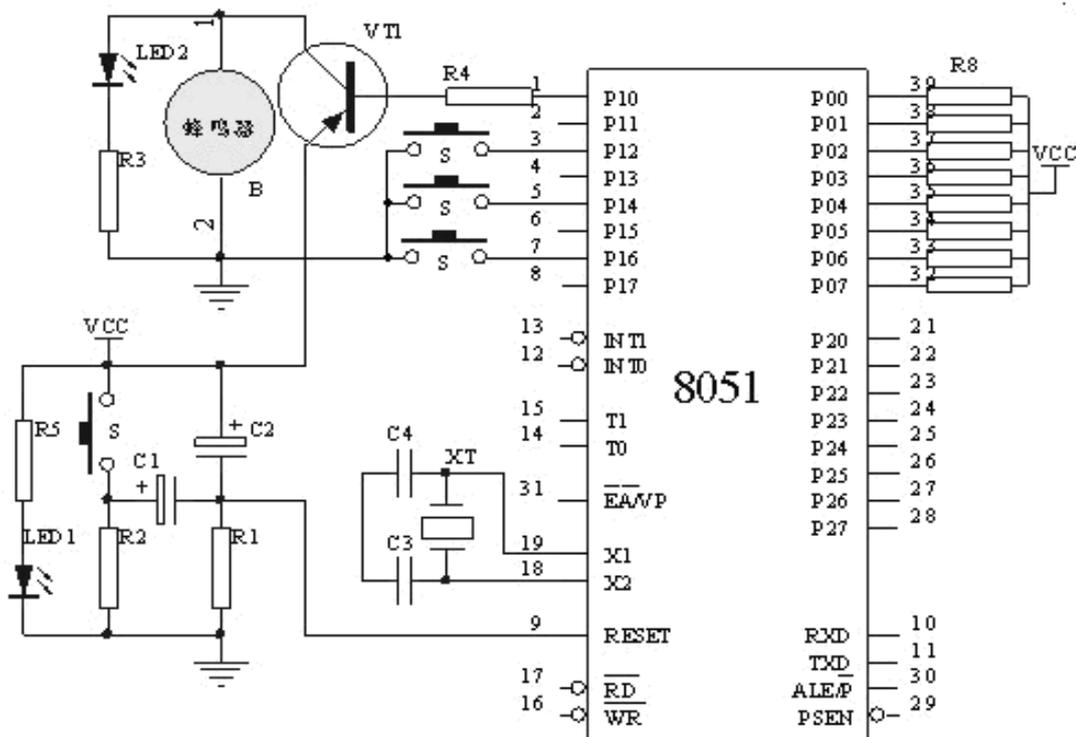


图 10 单片机主板电路

## 4.3 温度显示电路

显示电路是使用的串口显示，这种显示最大的优点就是使用口资源比较少，只用 p3 口的 RXD, 和 TXD, 串口的发送和接收，四只数码管采用 74LS164 右移寄存器驱动，显示比较清晰。

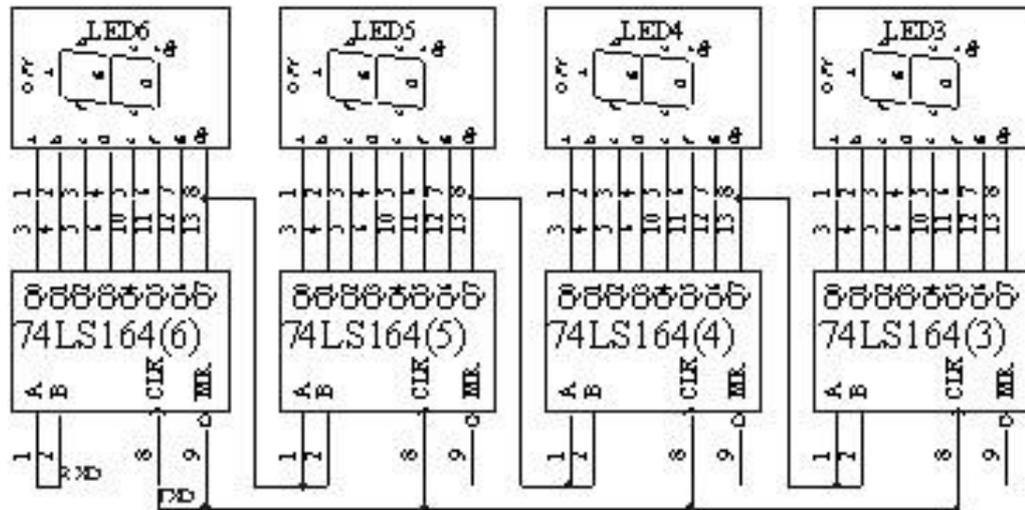


图 11 温度显示电路

## 5 总结

经过这次单片机毕业设计，终于完成了我的智能温度报警器的设计，虽然没有完全达到理想的设计要求，但从心底里说，还是高兴的。

在本次设计的过程中，我发现很多的问题，这次设计真的让我长进了很多，单片机课程设计重点就在于软件算法的设计，需要有很巧妙的程序算法，虽然以前写过几次程序，但我觉的写好一个程序并不是一件简单的事。从这次的毕业设计中，我真真正正的意识到，在以后的学习中，要理论联系实际，把我们所学的理论知识用到实际当中，学习单片机更是如此，程序只有在经常的写与读的过程中才能提高，这就是我在这次毕业设计中的最大收获。

### 【参考文献】

- [1] 李朝青. 单片机原理及接口技术（简明修订版）[M]. 北京航空航天大学出版社:1998, 11-15
- [2] 李广弟. 单片机基础[M]. 北京航空航天大学出版社:1994, 23-25
- [3] 陈京培, 徐永梅. 基于AT89S52单片机的液晶显示[J]. 现代电子技术, 2008, 31 (22) :22-25.
- [4] 高云红. 数字温度传感器在多点温度测量系统中的应用[J]. 沈阳航空工业学院学报, 2006, 23 (2) :61-63

## 附录 A（程序清单）

程序采用单片机汇编语言编写程序如下：

\*\*\*\*\*定义常数\*\*\*\*\*

```
TIMEL      EQU      0E0H ;           延时20ms
TIMEH      EQU      0B1H
TEMPHEAD   EQU      36H
```

\*\*\*\*\*定义工作内存\*\*\*\*\*

```
BITST      DATA    20H
TIME1SOK   BIT      BITST.1
TIMPONEOK  BIT      BITST.2
TEMPL      DATA    26H
TEMPH      DATA    27H
TEMPHC     DATA    28H
TEMPLC     DATA    29H
```

\*\*\*\*\*DS18B20数据输入端口定义\*\*\*\*\*

```
TEMPDIN    BIT      P3.7
            ORG      0000H
            LJMP    START
            ORG      00BH
            LJMP    TOIT
```

\*\*\*\*\*初始化\*\*\*\*\*

```
            ORG      100H
START:      MOV      SP, #06H
CLSMEM:     MOV      R0, #20H
            MOV      R1, #60H
CLSMEM1:    MOV      @R0, 00H
            INC      R0
            DJNZ    R1, CLSMEM1
            MOV      TMOD, #00100001B;   定义定时器0工作方式（16位）；
            MOV      TH0, #TIMEL
            MOV      TLO, #TIMEH
            SJMP    INIT
ERROR:      NOP
            LJMP    START
            NOP
INIT:       NOP
            SETB    ETO
            SETB    TRO
            SETB    EA
```

```

MOV     PSW, #00H
CLR     TEMPONEOKL
LJMP    MNIN
;*****定时器0中断程序*****
TOIT:   PUSH    PSW
        MOV     PSW, #10H
        MOV     TH0, #TIMEH
        MOV     TLO, #TIMEL
        INC     R7
        CJNE    R7, #32H, TOIT1
        MOV     R7, #00H
        SETB    TIME1SOK
TOIT1:  POP     PSW
        RETI
*****主程序*****
MAIN:   LCALL   DISP1;           调用显示子程序
        JNB    TIME1SOK, MAIN
        CLR    TIME1SOK;        测温问1S一次
        JNB    TEMPONEOK, MAIN2; 上电时先温度转换
        LCALL  READTEMP1;       读出温度子程序
        LCALL  CONVTEMP;        温度BCD码计算处理子程序
        LCALL  DISPBCD;         BCD码刷新
        LCALL  DISP1;           消闪烁，显示一次
MAIN2:  LCALL  READTEMP;        温度转换开始
        SETB   TEMPONEOK
*****子程序*****
INITDS1820: SETB    TEMPDIN
          NOP
          NOP
          CLR    TEMPDIN
          MOV    R6, #0A0H;      延时480 μs
          DJNZ   R6, $
          MOV    R6, #0A0H
          DJNZ   R6, $
          SETB   TEMPDIN
          MOV    R6, #32
          DJNZ   R6, $
          MOV    R6, #3CH
LOOP1820: MOV     C, TEMPDIN
          JC     INITDS1820OUT

```

```

        DJNZ    R6, #LOOP1820
        MOV     R6, #064H
        DJNZ    R6, $
        SJMP    INITDS1820
        RET
INITDS1820OUT: SETB    TEMPIN
        RET
*****读DS18B20的程序，读出一字节的数据*****
READDS1820:   MOV     R7, #08H
        SETB    TEMPIN
        NOP
        NOP
READDS1820LOOP: CLR     TEMPIN
        NOP
        NOP
        NOP
        SETB    TEMPIN
        MOV     R6, #07H
        DJNZ    R6, $
        MOV     C, TEMPIN
        MOV     R6, #3CH
        DJNZ    R6, $
        RRC     A
        SETB    TEMPIN
        DJNZ    R7, READDS1820LOOP
        MOV     R6, #3CH
        DJNZ    R6, $
        RET
*****写DS18B20程序*****
WRITEDS1820:  MOV     R7, #08H
        SETB    TEMPIN
        NOP
        NOP
WRTIEDS1820LOP: CLR     TEMPIN
        MOV     R6, #07H
        DJNZ    R6, $
        RRC     A
        MOV     TEMPIN, C
        MOV     R6, #34H
        DJNZ    R6, $
    
```

```

        SETB     TEMPDIN
        DJNZ    R7, WRITEDS1820LOP
        RET
*****读 TEMP*****
READTEMP:    LCALL    INITDS1820
             MOV     A, #0CCH
             LCALL    WRITEDS1820
             MOV     R6, #34H
             DJNZ    R6, $
             MOV     A, #44H
             LCALL    WRITEDS1820
             MOV     R6, #34H
             DJNZ    R6, $
             RET
READTEMP1:   LCALL    INITDS1820
             MOV     A, #0CCH
             LCALL    WRITEDS1820
             MOV     A, #34H
             DJNZ    R6, $
             MOV     A, #0BEH
             LCALL    WRITEDS1820
             MOV     R6, #34H
             DJNZ    R6, $
             MOV     R5, #09H
             MOV     R0, #TEMPEAD
             MOV     B, #00H
READTEMP2:   LCALL    READDS1820
             MOV     @R0, A
             INC     R0
READTEMP21:  LCALL    CRC8CAL
             DJNZ    R5, READTEMP2
             MOV     A, B
             JNZ     REATDTEMPOUT
             MOV     A, TEMPHEAD+0
             MOV     TEMPL, A
             MOV     A, TEMPHEAD+1
             MOV     TEMPH, A
READTEMPOUT: RET
*****处理温度BCD码子程序*****
CONVIEMP:   MOV     A, TEMPH
    
```

```

ANL      A, #80H
JZ       TEMPC1
CLR      C
MOV      A, TEMPL
CPL      A
ADD      A, #01H
MOV      TEMPL, A
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      TEMPHC, A
MOV      A, TEMPL
ANL      A, #0FH
MOV      DPTR, #TEMPDOTTAB
MOVC     A, @A + DPTE
MOV      TEMPLC, A
MOV      A, TEMPL
ANL      A, #0FOH
SWAP     A
MOV      TEMPL, A
MOV      A, TEMPL
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, TEMPPLC

```

```

MOV     TEMPPLC, 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
*****小数部分码表*****
TEMPDOTTAB:  DB      00H, 01H, 01H, 02H, 03H, 04H, 04H, 05H, 06H
              DB      06H, 07H, 08H, 08H, 09H, 09H
*****显示BCD码刷新子程序*****
DISPBCD:  MOV     A, TEMPLC
          ANL     A, #0FH
          MOV     70H, A
          MOV     A, TEMPLC
          SWAP    A
          ANL     A, #0FH
          MOV     72H, A
          MOV     A, TEMPHC
          SWAP    A
          ANL     A, #0FH
          MOV     A, TEMPHC
          ANL     A, #0FOH
          CJNE   A, #010H, DISPBCD0
          SJMP   DISPBCD2
DISPBCD0: MOV     A, TEMPHC
          ANL     A, #0FH
          JNZ    DISPBCD2
          MOV     A, TEMPHC
          SWAP    A
          ANL     A, #0FH
          MOV     73H, #0AH
          MOV     72H, A
DISPBCD2:  RET
*****显示子程序*****
P1口输出   在70H--73H内   P3扫描控制

```

```

DISP1:      MOV      R1, #70H
            MOV      R5, #0FEH
PALY:       MOV      P1, #0FFH
            MOV      A, R5
            MOV      P3, A
            MOV      A, @R1
            MOV      DPTR, #TAB
            MOVC     A, @A +DPTR
            MOV      P1, A
            JB       ACC. 1, LOOP5
            CLR      P1. 7
LOOP5:      LCALL   DL1MS
            INC      R1
            MOV      A, R5
            JNB     ACC. 3, ENDOUT
            RL       A
            MOV      R5, A
            AJMP    PALY
ENDOUT:     MOV      P1, #0FFH
            MOV      P3, #0FFH
            RET
TAB:        DB
            0C0H, 0F9H, 0A4H, 0B0H, 99H, 92H, 82H, 0F8H, 80H, 90H, 0FFH, 0BFH
DL1MS:      MOV      R6, #14H
DL1:        MOV      R7, #19H
DL2:        DJNZ    R7, DL2
            DJNZ    R6, DL1
            RET
*****转换BCD*****
HEX2BCD1:   MOV      B, #064H
            DIV     AB
            MOV      R7, A
            MOV      A, #0AH
            XCH     A, B
            DIV     AB
            SWAP    A
            ORL     A, B
            RET
*****CRC——8校验****
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
           END
    
```