源程序代码:

```
        FLAG1   BIT   F0              ;DS18B20存在标志位
        DQ BIT P3.2                   ;读DS18B20的数据端
        TEMPER_L EQU 29H              ;存温度的低字节
        TEMPER_H EQU 28H              ;存温度的高字节
        A_BIT EQU 35H                 ;存正数部份的个位
        B_BIT EQU 36H                 ;存正数部份的十位
        ZJJ EQU 37H                   ;过度位
        XSB EQU  38H                  ;过度位
        ORG 0000H                     ; DS18B20汇编程序起始
        AJMP MAIN                     ;跳到主程序
        ORG 0100H
MAIN:
        LCALL INIT_18B20              ;调用复位时序
        LCALL RE_CONFIG               ;调用设置精度子程序
        LCALL GET_TEMPER              ;调用获取温度数值程序
        AJMP CHANGE                   ;调用温度转换成BCD码的程序
INIT_18B20:
        SETB DQ                       ;DQ位置1
        NOP
        CLR DQ                        ;拉低DQ电平
        MOV R0,#0FBH                  ;设置延长的时间具体是多少
TSR1:
        DJNZ R0,TSR1                  ;延时
        SETB DQ                       ;DQ位置1
        MOV R0,#25H                   ;设置延长的时间具体是多少
TSR2:
        JNB DQ ,TSR3                   ;看看DQ位是否为0，如果为0则说明
```

```
                                ;DS18B20不存在则顺序执行
        DJNZ R0,TSR2            ;延时
TSR3:
        SETB FLAG1              ;置标志位，表明DS18B20存在
        CLR P3.7                ;二极管指示
        AJMP TSR5               ;存在的情况下跳到TSR5，否则顺序执行
TSR4:
        CLR FLAG1
        LJMP TSR7
TSR5:
        MOV R0,#06BH            ;要等待的具体时间
TSR6:
        DJNZ R0,TSR6            ;延时
TSR7:
        SETB DQ                 ;表明不存在
        RET
RE_CONFIG:
        JB FLAG1,RE_CONFIG1
        RET
RE_CONFIG1:
        MOV A,#0CCH             ;放跳过ROM命令
        LCALL WRITE_18B20
        MOV A,#4EH              ;写暂存器命令
        LCALL WRITE_18B20       ;写暂存器命令
        MOV A,#00H              ;报警上限中写入00H
        LCALL WRITE_18B20       ;调用写DS18B20一个字节命令
        MOV A,#00H              ;报警下限中写入00H
        LCALL WRITE_18B20       ;调用写DS18B20一个字节命令
        MOV A,#7FH              ;选择十二位温度分辨率
```

```
        LCALL WRITE_18B20              ;调用写DS18B20一个字节命令
        RET
GET_TEMPER:
        SETB DQ
        LCALL INIT_18B20
        JB FLAG1,TSS2
        RET                            ;若不存在则返回
TSS2:
        MOV A,#0CCH                    ;跳过ROM
        LCALL WRITE_18B20              ;调用写DS18B20一个字节命令
        MOV A,#44H                     ;发出温度转换命令
        LCALL WRITE_18B20              ;调用写DS18B20一个字节命令
        LCALL DISPLAY                  ;延时
        LCALL INIT_18B20               ;调用复位时序
        MOV A,#0CCH ;跳过ROM
        LCALL WRITE_18B20              ;调用写DS18B20一个字节命令
        MOV A,#0BEH                    ;发出读温度换命令
        LCALL WRITE_18B20              ;调用写DS18B20一个字节命令
        LCALL READ2_18B20              ;读两个字节的温度
        RET
WRITE_18B20:
        MOV R2,#8                  ;一个字节8位
        CLR C                      ;进位位清0
WR1:
        CLR DQ                 ;拉低DQ的电平
        MOV R3,#6              ;等待的具体时间
        DJNZ R3,$              ;延时
        RRC A                  ;累加器A带进位位右移
        MOV DQ,C               ;C的值给DQ
```

```
        MOV R3,#23              ;等待的具体时间
        DJNZ R3,$               ;延时
        SETB DQ                 ;拉高DQ的电平
        NOP
        DJNZ R2,WR1             ; 延时
        SETB DQ                 ;拉高DQ的电平
        RET
READ2_18B20:
        MOV R4,#2               ;低位存在29H,高位存在28H
        MOV R1,#29H
RE00:
        MOV R2,#8               ;一个字节8位
RE01:
        CLR C
        SETB C
        NOP
        NOP
        CLR DQ                  ;拉低DQ的电平
        NOP
        NOP
        NOP
        SETB DQ
        MOV R3,#7               ;等待的具体时间
        DJNZ R3,$               ;延时
        MOV C,DQ                ;DQ的值给C
        MOV R3,#23              ;等待的具体时间
        DJNZ R3,$               ;延时
        RRC A                   ;累加器A 带进位位右移
        DJNZ R2,RE01            ;两个字节读完，否则继续读
```

```
        MOV @R1,A                   ;把读的字节给29H ,28H
        DEC R1
        DJNZ R4,RE00
        RET
CHANGE:
        MOV A,29H
        MOV R5,29H
        MOV C,28H.0                 ;将28H中的最低位移入C
        RRC A
        MOV C,28H.1
        RRC A
        MOV C,28H.2
        RRC A
        MOV C,28H.3
        RRC A
        MOV 29H,A                   ;这一小段程序是把28H的低四位与29H
                                    ;的高四位组合一个新字节
        MOV A,R5
        ANL A,#00001111B            ;把29H的低四位拿出来考虑
        MOV B,#8
        DIV AB                      ;得到的数据除以8
        MOV B,#5
        MUL AB                      ;之后得到的数据乘以5
        MOV 38H,A                   ;这一小段程序则是按论文的要求小数
                                    ;点后面的精度为0.5
        LCALL DISPLAY               ;调用数码管显示子程序
        LJMP MAIN
DISPLAY:
        MOV A,29H                   ;将29H中的十六进制数转换成10进制
```

```
        MOV B,#100
        DIV AB                  ;除以100把百位数据除掉剩下个位与
                                ;十位，这条指令可要可不要
        MOV A,B                 ;把余数给 A
        MOV B,#10
        DIV AB                  ;所得的数除以10
        MOV a_bit,A             ;十位在A
        MOV a_bit,B             ;个位在B
        MOV A,29H               ;把新组成的字节的数给A
        SUBB A,#33              ;把新组成的字节的数与33相减
        JC swcz                 ;如果Cy为1说明温度低于33度，要跳到
                                ;升温操作上去
        CLR C
        MOV A,29H               ;把新组成的字节的数给A
        SUBB A,#37              ;把新组成的字节的数与37相减
        JNC jwcz                ;如果Cy为0说明温度高于37度，要跳到
                                ;降温操作上去
        SETB P3.0               ;如果在33度与37度之间说明不用做任何
                                ;操作
        SETB P3.1
        LJMP 111
swcz:
        CLR  P3.1                ;升温操作P3.1复位
        SETB P3.0                ;升温操作P3.0置位
        LJMP 111
jwcz:
        CLR P3.0                  ;降温操作P3.0复位
        SETB P3.1                 ;降温操作P3.1置位
```

```
lll:
        MOV DPTR,#TAB                   ;指定查表启始地址
        MOV R0,#10
dpl1:
        MOV R1,#120                     ;设置循环的次数
dplop:
        MOV A,38H                       ;取小数位数
        MOVC A,@A+DPTR                  ;查个位数的7段代码
        MOV P1,A                        ;送出小数的7段代码
        MOV P2,#00000100B               ;开小数位显示
        ACALL D1MS                      ;显示1ms
        MOV P2,#00000000B               ;关小数位显示
        MOV A,a_bit                     ;取个位数
        MOVC A,@A+DPTR                  ;查个位数的7段代码
    ADD A,#80h                      ;使个位后有小数点
    MOV P1,A                        ;送出个位的7段代码
    MOV P2,#00000010B               ;开个位显示
    ACALL D1MS                      ;显示1ms
    MOV P2,#00000000b               ;关个位显示
    MOV DPTR,#TAB
    MOV A,b_bit                     ;取十位数
    MOVC A,@A+DPTR                  ;查十位数的7段代码
    MOV P1,A                        ;送出十位的7段代码
    MOV P2,#00001000B               ;开十位显示
    ACALL  D1MS                     ;显示1ms
    MOV P2,#00000000B               ;关十位显示
    DJNZ R1, dplop
    DJNZ R0,dpl1                    ;1200次没循环完
    RET
```

```
D1MS:   MOV R7,#80                    ;1MS延时(按12MHZ算)
        DJNZ R7,$
        RET
TAB:    DB 3FH,06H,5BH,4FH,66H,6DH,07DH,07H,7FH,6FH
        END
```