

```

    ORG 0000H
    AJMP MAIN
    ORG 0013H
    AJMP INT0

MAIN:
    ORG 0100H
    ;初始化
    MOV 30H, #0H
    MOV 31H, #0H
    MOV 32H, #0H
    MOV 33H, #0H

    SETB EA
    SETB EX1
    SETB IT1                ;开中断

    ;第一次开启AD转换
    MOV DPTR, #0DFF8H      ;A13=0, 并指向IN0通道
    MOVX @DPTR, A          ;启动AD转换
    CLR P1.6

LOOP:
    MOV A, 40H             ;处理IN0, 得空气温度
    ;MOV R1, #32H          ;空气温度小数、个、十放置于30H, 31H, 32H, R1存放ADC数字量置数区
    ;CLR P1.6
    ACALL DELAY_LONG
    ACALL TUNBCD           ;在TUNBCD中处理A的数据

    ACALL DISPLAY          ;刷新数码管
    SJMP LOOP              ;死循环, 等待中断

    ORG 0200H

INT0:
    CLR EX1                ;中断发生时关闭中断使能
    PUSH PSW                ;现场保护, 中断地址压栈
    PUSH ACC                ;寄存器压栈
    MOV DPTR, #0DFF8H      ;选择IN0通道
    MOVX A, @DPTR           ;转换结束, 读入转换结果
    MOV 40H, A              ;将AD转换数放入40H中储存
    MOVX @DPTR, A           ;再次启动, 新一轮温度刷新
    POP ACC                 ;弹出累加器
    POP PSW                 ;断点地址入PC
    SETB EX1                ;重新开启中断使能
    RETI

;*****
;* 显示数据转为三位BCD码程序 *
;*****
;显示数据转为三位BCD码存入32H、31H、30H(最大值5.00v)
TUNBCD: ;255/51=5.00V运算
    MOV B, #51 ;
    DIV AB ;
    MOV 32H, A ;整数为十位, 放到第三个数码管

;*****以下处理余数*****
    MOV A, B ;余数大于19H, F0为1, 乘法溢出, 结果加5
    CLR F0
    SUBB A, #1AH
    MOV F0, C

```

```
    MOV A, #10 ;
    MUL AB ;
    MOV B, #51 ;
    DIV AB
    JB F0, LOOP2 ;
    ADD A, #5
LOOP2: MOV 31H, A ;个位，第二个数码管
    MOV A, B
    CLR F0
    SUBB A, #1AH
    MOV F0, C
    MOV A, #10 ;
    MUL AB ;
    MOV B, #51 ;
    DIV AB
    JB F0, LOOP3 ;
    ADD A, #5
LOOP3: MOV 30H, A ;小数位，第一个数码管
;*****余数处理完毕*****;

    RET
;
;*****数码管刷新*****;
DISPLAY:
    ;第一位
    MOV DPTR, #DSEG1
    MOV A, 30H
    MOVC A, @A+DPTR
    MOV DPTR, #7FF0H
    MOVX @DPTR, A

;    ACALL DELAY_SHORT

    ;第二位
    MOV DPTR, #DSEG1
    MOV A, 31H
    MOVC A, @A+DPTR
    MOV DPTR, #7FF1H
    CLR ACC.7
    MOVX @DPTR, A

;    ACALL DELAY_SHORT

    ;第三位
    MOV DPTR, #DSEG1
    MOV A, 32H
    MOVC A, @A+DPTR
    MOV DPTR, #7FF2H
    MOVX @DPTR, A

;    ACALL DELAY_SHORT

    ;第四位
    MOV DPTR, #DSEG1
    MOV A, 33H
    MOVC A, @A+DPTR
    MOV DPTR, #7FF3H
    MOVX @DPTR, A

;    ACALL DELAY_SHORT

    RET
```

DELAY_SHORT: ;延迟

```
    MOV R7,#10
DEL_SHORT: MOV R6,#250
           DJNZ R6,$
           DJNZ R7,DEL_SHORT
           RET
```

DELAY_LONG: ;延迟

```
    CLR EX1
    MOV R7,#250
DEL_LONG: MOV R6,#250
           DJNZ R6,$
           DJNZ R7,DEL_LONG
           SETB EX1
           RET
```

DSEG1: DB 0C0H,0F9H,0A4H,0B0H ;段码
DB 99H,92H,82H,0F8H
DB 80H,90H,88H,83H
DB 0C6H,0A1H,86H,8EH

END