杭州晶华微电子有限公司
Hangzhou SDIC Microelectronics Co.,Ltd.
浙江省杭州市滨江区长河路351号拓森科技园4号楼5楼
电话：0571-86673068, 86673071 传真：0571-86673072
电邮：info@SDICmicro.cn 网址: www.SDICmicro.cn

基于 SD8000T 的温度测量应用

作者：许为来

2016 年 12 月 30 日

版本： 0

目录

1. 简介

SD8000T 芯片内部自带温度测量功能，无需外接温度传感器即可进行温度测量，测量出来的温度为绝对温度，最大误差为 2℃，支持单点校准。

在进行温度校准时，需要配合我司提供的烧录器（1.86 及以上版本）进行使用，温度校准和温度测量的函数请调用我司提供的“TemperatureCAL.asm”文件。

2. 设计参考

2.1 温度校准

2.1.1 烧录器写入标准温度值

烧录器在进行温度校准前，会往芯片内部写入两个字节的温度值（高字节在前），温度值的换算方法如图 2.1 所示。

温度	温度测量结果（二进制）
-55℃	1100 1001 0000 XXXX
-40℃	1101 1000 0000 XXXX
-25℃	1110 0111 0000 XXXX
-0.0625℃	1111 1111 1111 XXXX
0	0000 0000 0000 XXXX
0.0625℃	0000 0000 0001 XXXX
25℃	0001 1001 0000 XXXX
75℃	0100 1011 0000 XXXX
80℃	0101 0000 0000 XXXX
100℃	0110 0100 0000 XXXX
125℃	0111 1101 0000 XXXX

图 2.1 温度值换算方法

烧录器会以自增的形式写入标准温度数据到芯片，例如本次校准时写入的地址是 0x1F80 和 0x1F81，那么下一次就会写入到 0x1F82 和 0x1F83，这样就可以实现多次校准，以最后一次写入的数据为准。

2.1.2 温度校准进入

上电时，将 P30 和 P31 口上拉，判断 P30 和 P31 口是否为 0，如果都为 0 则使 P33 口输出 0，然后调用“TemperatureCALControl”函数。

在“TemperatureCALControl”函数内部首先等待高压标志位，当有高压标志位后则进入温度校准。

2.1.3 温度校准

调用“Temperature_Measure”函数进行温度转换，该函数的流程描述如下。

- 初始化 ASP 模块
  - 开启 AVDDR、BG 和 ACM；
  - 关闭 Buffer 的 Chopper，开启 Buffer；
  - 关闭 PGIA 或设置为 1 倍放大，关闭 PGIA 的 Chopper；
  - 基准设置为 0.6V；
  - 开启 ADC。

- 采集温度数据
  - 设置 ADC 差分采样通道为 Tsensorp 和 Tsensorn（正向）；
  - 读取第 5 笔转换数据（SINC3 滤波器），存储为 Ta；
  - 设置 ADC 差分采样通道为 ACM（反向）；
  - 读取第 5 笔转换数据（SINC3 滤波器），存储为 Tb；
  - TD=Ta-Tb 即得到一次温度测量数据；
  - 温度采集完成后，若数据不正确（小于 0x0600）,则置错误标志。

温度采集完成后，判断是否采集正确，如果采集错误，则置 P30 为 1，然后置 P33 为 1，温度校准正确时，则将温度数据写入芯片内部（一共两个字节，高字节在前）并进行校验，如果烧录和校验都正确，则置 P30 为 0，然后置 P33 为 1，否则置 P30 为 1，然后置 P33 为 1。

芯片在写入温度测量数据时，假如是从 0x1F00 地址开始写入的，程序会自动以自增地址的形式向下两个地址进行写入，这样就可以完成多次的校准，以最后一次校准的数据为准。

当烧录器判断到 P33 置为 1 后，将判断 P30 口的电平，从而可以判断是否校准正确。

3. 温度测量

如果没有进行校准，程序会默认使用一个系数进行计算温度，该系数可设置为 0x0618。如果已经进行校准，上电后，可以通过读取第 2 节描述的烧录器写入的标准温度值和芯片自行写入的温度测量数据进行计算，计算公式如式 3-1 所示。

$$C = Tref \times 1000 / TDref \qquad （式 3-1）$$

其中 C 为最终的校准系数（放大了 1000 倍），Tref 为烧录器写入的标准温度值，TDref 为芯片测量的温度数据（ADC 转换的结果）。

3.1 测量方法

调用“Temperature_Measure”函数可进行温度测量，将测量的结果按照式 3-2 进行运算即可得到实际的温度值。

$$Treal = C \times TDreal / 1000 \qquad （式 3-2）$$

其中 C 为校准系数，Treal 为实际的温度值，TDreal 为当前测量的温度数据。

3.2 流程图

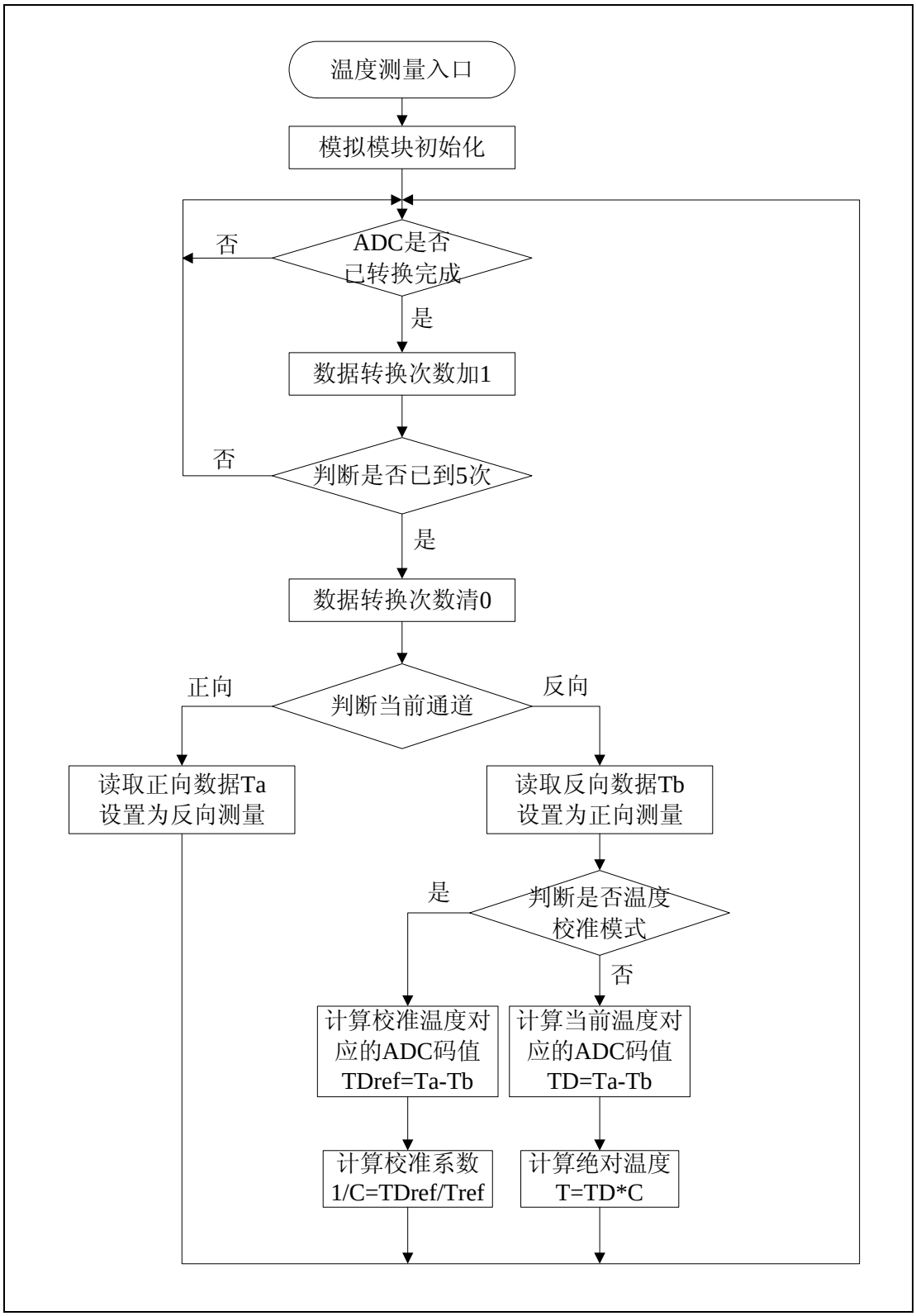

图 3.1 测温流程图

3.3 示例代码

```
        bsf         TemperatureCal          ;置温度校准标志位

        movlw       0x93
        movwf       AVDDR,0                 ;模拟部分 2.4V, 数字部分 2V

        movlw       0x0E
        movwf       DFM,0                   ;SINC3, 8192 过采样率

        movlw       0xFC
        movwf       ASPCK1,0                ;设置 ADC 时钟为 125kHz

        movlw       0x14
        movwf       ASPMUXS1,0              ;设置基准为 0.6V, Tsensor p / n

        movlw       0x01
        movwf       ASPMUXS3,0              ;AIP-AIN

        movlw       0xE1                    ; 1 倍增益,关 PGIA,BUFON
        movwf       ASPM,0
                                            ;ADC 使能,chopperoff
        clrf        count,1                 ;计数器清 0

    LoopGetTemp:
        btfss       PIR3,ADIF,0             ;等待 ADC 转换完成标志产生
        bra         LoopGetTemp

        bcf         PIR3,ADIF,0
        incf        count,1,1

        movlw       .5                      ;读取第 5 笔数据
        cpfseq      count,1
        bra         LoopGetTemp

        clrf        count,1

        movlw       0x14                    ;判断测量方向
        cpfseq      ASPMUXS1,0
        bra         LoopGetTempReverse
```

```
    ;正向
    movf        ADCDH,0,0               ;读取正向测量数据
    movwf       temp + 0,1
    movf        ADCDU,0,0
    movwf       temp + 1,1

    movlw       0x13                    ;设置为反向测量
    movwf       ASPMUXS1,0

    bra         LoopGetTemp

    ;反向
LoopGetTempReverse:
    movf        ADCDH,0,0               ;正向减去反向测量数据
    subwf       temp + 0,1
    movf        ADCDU,0,0
    subwfb      temp + 1,1

    movlw       0x14                    ;设置为正向测量
    movwf       ASPMUXS3,0

    btfsc       TemperatureCal
    bra         TemperatureCalMode

    ;无温度校准标志，计算温度
    movf        temp + 0,0,1
    movwf       temperature_code_l,1
    movf        temp + 1,0,1
    movwf       temperature_code_h,1

    clrf        divdend_U,1
    clrf        divdend_HH,1
    clrf        divdend_HL,1
    movf        temperature_code_h,0,1
    movwf       divdend_LH,1
    movf        temperature_code_l,0,1
    movwf       divdend_LL,1

    clrf        divsor_HH,1
    clrf        divsor_HL,1
    movf        temperature_quotiety_h,0,1
    movwf       divsor_LH,1
    movf        temperature_quotiety_l,0,1
    movwf       divsor_LL,1
```

```
    call        div_process             ;调用乘法子函数

    movf        quotient_LL,0,1         ;保存绝对温度值
    movwf       temperature_l,1
    movf        quotient_LH,0,1
    movwf       temperature_h,1

    bra         LoopGetTemp

    ;有温度校准标志，计算校准系数
TemperatureCalMode:
    bcf         TemperatureCal          ; 清温度校准标志位

    movf        temp + 0,0,1
    movwf       cal_temperature_code_l,1
    movf        temp + 1,0,1
    movwf       cal_temperature_code_h,1

    ;校准温度在程序烧录过程中已写入 OTP，可从 OTP 中读取
    ;保存在 cal_temperature_h 和 cal_temperature_l 寄存器中
    clrf        divdend_U,1
    clrf        divdend_HH,1
    clrf        divdend_HL,1
    movf        cal_temperature_code_h,0,1
    movwf       divdend_LH,1
    movf        cal_temperature_code_l,0,1
    movwf       divdend_LL,1

    clrf        divsor_HH,1
    clrf        divsor_HL,1
    movf        cal_temperature_h,0,1
    movwf       divsor_LH,1
    movf        cal_temperature_l,0,1
    movwf       divsor_LL,1

    call        div_process             ;调用除法子函数

    movf        quotient_LL,0,1         ;保存温度系数 1/C
    movwf       temperature_quotiety_l,1
    movf        quotient_LH,0,1
    movwf       temperature_quotiety_h,1

    bra         LoopGetTemp
```