



产品规格承认书

客户名称:

产品型号: STP10DB51G6

规格书编号:

产品版本:

日期:

本公司确认			
制作	工程	审核	批准

客户回签栏:

上海烨映微电子科技股份有限公司
上海市桂平路481号16幢5层
Tel: 021-64263818
Fax: 021-51862418
E-mail: info@yysensor.com

变更记录

序号	更改内容	更改原因	更改时间	编制
1	格式调整		2024.9.2	彭晓占

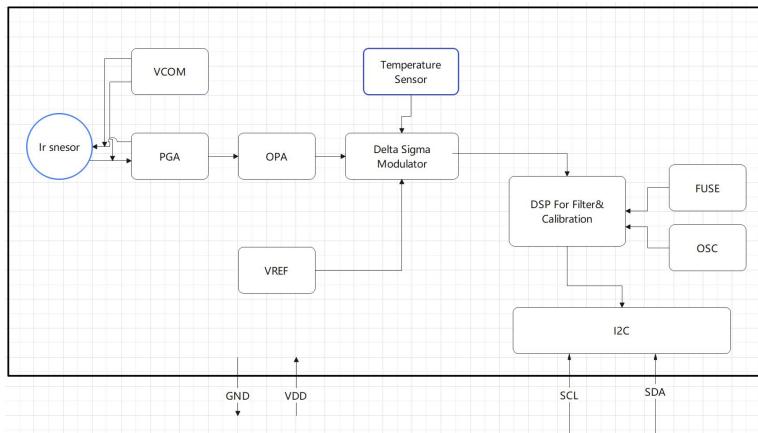
目录

No.	内容	页码
1	封面	1
2	变更记录	2
3	目录	3
4	电气特性	4
5	产品方框图	5
6	机械尺寸图	5
7	引脚定义	5
8	光学特性	6
9	温度性能图表	6-7
10	可靠性测试	7-8
11	主要物料清单	8
12	产品包装及储存	9
13	注意事项	9
14	通讯协议	10-18
15	样品检验报告	19

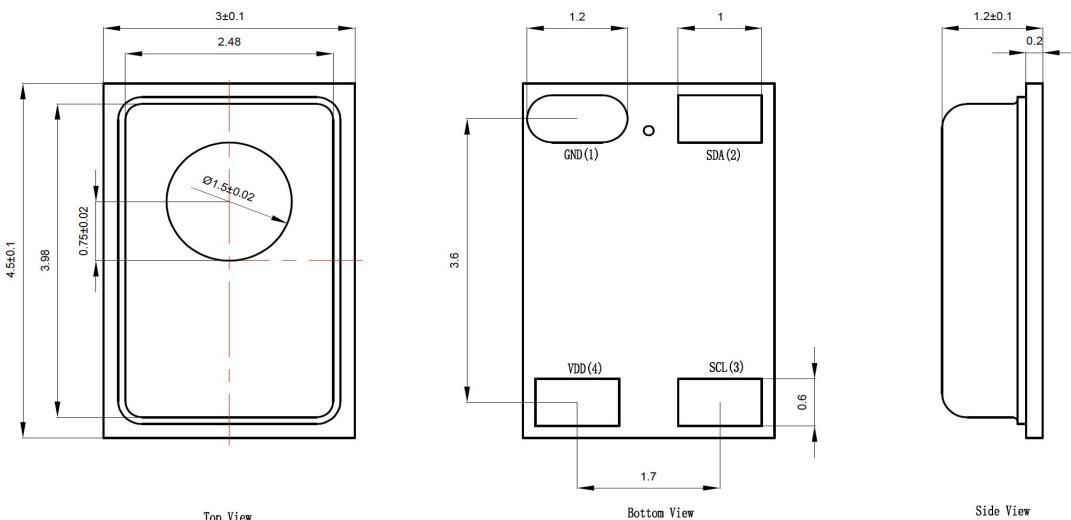
电气特性

参数	条件	最小值	典型值	最大值	单位
供电电压		3.0	3.3	5.5	V
消耗电流	工作模式@3.3V			1.5	mA
	IDLE@3.3V			1	uA
测温范围		-20		150	°C
测温分辨率			0.1		°C
视场角			120		°
通讯速度				400	kHz
通讯地址			0x58		
温度数据刷新频率			1		Hz
工作温度		-20		105	°C
存储温度		-40		125	°C
精度	PTAT=25°C Vobj=20~100°C		±3		°C

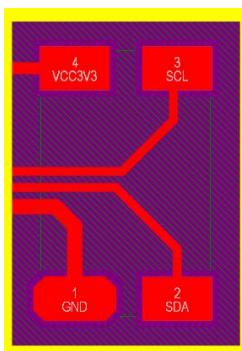
产品方框图



机械尺寸图 (单位mm)



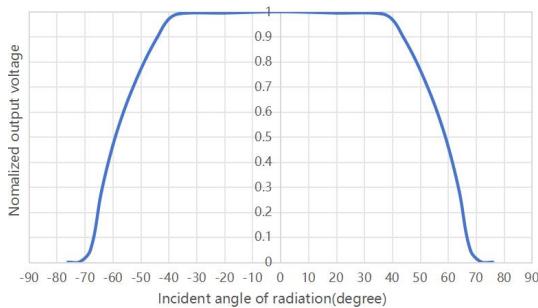
引脚定义



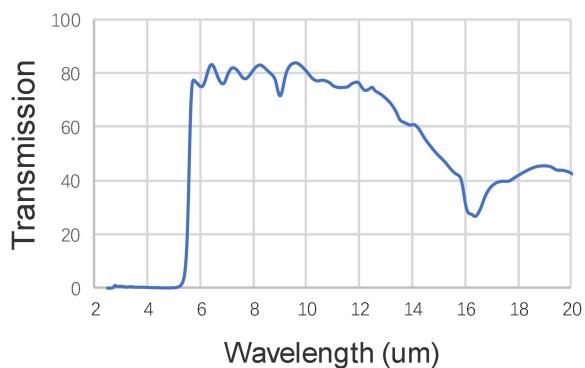
引脚序号	引脚定义	功能描述
1	GND	电源-
2	SDA	IIC数据输入输出
3	SCL	IIC时钟输入输出
4	VDD	电源+

光学特性

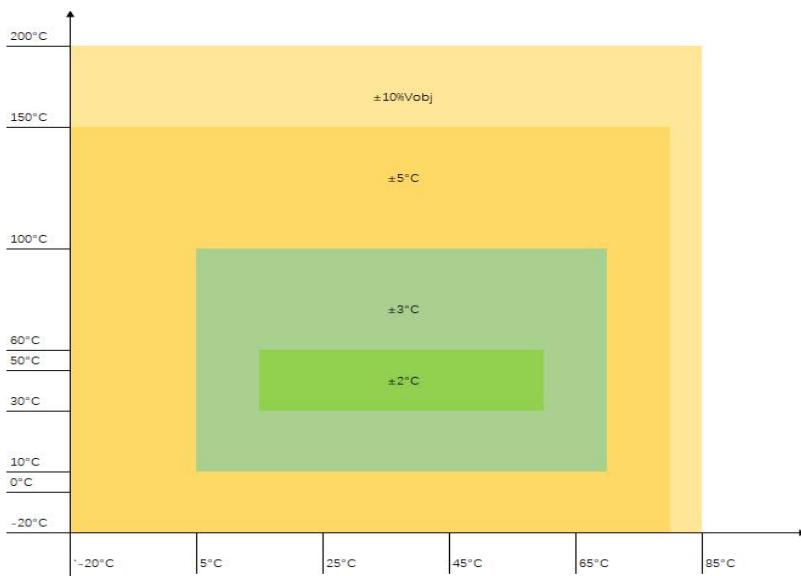
FOV特性



红外透光波长曲线



温度性能图表



性能图表

- 当环境温度高达105°C时，传感器功能仍可工作。但是，如果环境温度超过80°C，则测温精度会有所下降；
- 所有精度规格仅适用于确定的等温条件。此外，只有当被测物完全填满传感器的视场角时，该精度才有效；
- 测试条件：
 - 传感器到黑体距离：3cm
 - 黑体大小：直径20厘米 黑体发射率:0.96

可靠性测试

NO.	项目	技术要求	测试条件	测试设备
1	低温存储	按规定的方 法试验，试 验后应符合 外观和检测 精度要求	将传感器一排（5个横向排列）放置在温箱内，开启环境箱，环境箱温度调到-40°C±2°C，放置168h后，取出常温下放置2h，读取传感器输出的环境温度(1秒为间隔，取60次)，传感器输出温度精度应满足±3°C，结构无变形等影响性能现象。 在室温下（22°C~28°C）把试验后的5个传感器固定于黑体测试仪前，根据传感器规格书标明的校准距离设置传感器与黑体的距离，分别设置黑体参考温度为35°C和50°C，读取传感器输出的目标温度(1秒为间隔，取60次)，温度检测精度应满足±3°C。	高低温箱
2	低温工作	按规定的方 法试验，试 验后应符合 外观和检测 精度要求	将传感器一排（5个横向排列）固定在环境箱内，开启环境箱，环境箱温度调到-20°C±2°C，传感器上电在额定电压工作168h，取出常温下放置2h，读取传感器输出的环境温度(1秒为间隔，取60次)，传感器输出温度精度应满足±3°C，结构无变形等影响性能现象。 在室温下（22°C~28°C）把试验后的5个传感器固定于黑体测试仪前，根据传感器规格书标明的校准距离设置传感器与黑体的距离，分别设置黑体参考温度为35°C和50°C，读取传感器输出的目标温度(1秒为间隔，取60次)，温度检测精度应满足±3°C。	高低温箱
3	高温存储	按规定的方 法试验，试 验后应符合 外观和检测 精度要求	将传感器一排（5个横向排列）放置在温箱内，开启环境箱，环境箱温度调到+105°C±2°C，放置48h后，取出常温下放置2h，读取传感器输出的环境温度(1秒为间隔，取60次)，传感器输出温度精度应满足±3°C，结构无变形等影响性能现象。 在室温下（22°C~28°C）把试验后的5个传感器固定于黑体测试仪前，根据传感器规格书标明的校准距离设置传感器与黑体的距离，分别设置黑体参考温度为35°C和50°C，读取传感器输出的目标温度(1秒为间隔，取60次)，温度检测精度应满足±3°C。	高低温箱
4	高温高湿工作	按规定的方 法试验，试 验后应符合 外观和检测 精度要求	将传感器一排（5个横向排列）固定在环境箱内，开启环境箱，环境箱温度调到+85°C±2°C、湿度85%RH±3%RH环境中，额定电压工作480h后，取出常温下放置2h，读取传感器输出的环境温度(1秒为间隔，取60次)，传感器输出温度精度应满足±3°C，结构无变形等影响性能现象。 在室温下（22°C~28°C）把试验后的5个传感器固定于黑体测试仪前，根据传感器规格书标明的校准距离设置传感器与黑体的距离，分别设置黑体参考温度为35°C和50°C，读取传感器输出的目标温度(1秒为间隔，取60次)，温度检测精度应满足±3°C。	恒温恒湿箱

可靠性测试

NO.	项目	技术要求	测试条件	测试设备
5	冷热冲击	按规定的方 法试验，试 验后应符合 外观和检测 精度要求。	按如下的顺序：5个传感器，在85°C±2°C环境中保持30min,然后立即放置(3min内)在-20°C±3°C的环境中30min,如此循环100次。取出后在常温下恢复2h, 读取传感器输出的环境温度(1秒为间隔, 取60次), 传感器输出温度精度满足±3°C, 结构无变形等影响性能现象。 在室温下(22°C~28°C)把试验后的5个传感器固定于黑体测试仪前, 根据传感器规格书标明的校准距离设置传感器与黑体的距离, 分别设置黑体参考温度为35°C和50°C, 读取传感器输出的目标温度(1秒为间隔, 取60次), 温度检测精度应满足±3°C。	冷热冲击箱
6	湿热交变	按规定的方 法试验，试 验后应符合 外观和检测 精度要求。	本试验按照GB-T 2423.34-2008 试验Z/AD, 参考20个循环的严酷等级进行试验。判定标准为480h内所有样品应功能正常，并无焊点裂纹、腐蚀、生长枝晶或其他缺陷。分别设置黑体参考温度为35°C和50°C, 读取传感器输出的目标温度(1秒为间隔, 取60次), 温度检测精度应满足±3°C。(输入电压：以设计典型值的上下20%为试验电压, 最低输入电压应大于4.5V, 高压与低压以1天为周期交替变化。)	湿热交变箱
7	盐雾试验	按规定的方 法盐雾，盐 雾试验后腐 蚀面积小于 3%，没有生 红锈。	按GB/T2423.17-93进行，放置在温度为35°C的盐雾箱内用浓度为5%氯化钠溶液喷雾24小时，试验后用水清洗干净。表面不允许有红锈生成。	盐雾试验箱

主要物料清单

No.	描述	供应商	供应商编码
1	电堆芯片 V3	烨映	/
2	ADC转换芯片	烨映	/

产品包装及储存

产品包装

No.	包装方式	包装材料/尺寸	数量
1	编带	12*16*24000mm	3000 pcs
2	包装箱	370*340*50mm	3000 pcs

储存:

储存温度:-10°C~40°C

相对湿度: RH60%以下

远离腐蚀和阳光照射

储存时间: 0.5年

使用注意事项

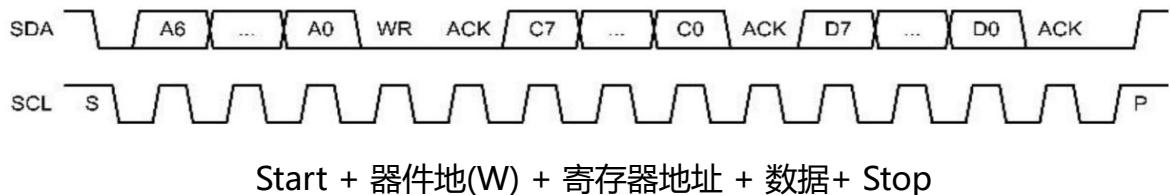
- 结构安装:** 传感器安装于设备时, 请确保视场覆盖所有检测区域, 红外传输滤光片应保持清洁, 无结露、污垢和划痕。使用光学部件和传感器组装时, 请考虑透射率、反射率、吸收率和温度分布。
- 物理冲击:** 当通过推、压、紧固或插入方式安装传感器时, 避免对传感器施加超过限制的振动、冲击和压力, 超限制的外力会导致微型热电堆结构破损和传感器失效。
- 引线处理:** 不要折叠引线基部的导线, 不要对引线施加等于或大于5N的拉力。
- 焊接限制:** 焊接和烙铁应仅与引线部分接触, 且在260°C以下条件下焊接时间不超过5秒。
- 助焊剂:** 使用含氯量低于0.2wt%的松香助焊剂, 并在焊接后彻底清洗, 以防止腐蚀。
- 清洁连接:** 使用乙醇清洗传感器光学镜面。在清洁无污垢、锈迹的情况下确认引线连接部分的连接情况, 避免松动。
- 有害环境:** 不要在腐蚀性气体(如氯气、氨气、二氧化硫、氮氧化物等)、高导电条件(如电解质、水、盐水)或粉尘多的环境中使用传感器。

通讯协议

IIC通讯的7位器件地址为0x58。

支持IIC的通讯标准速率是100K和IIC的高速速率400K

写时序：



解析：

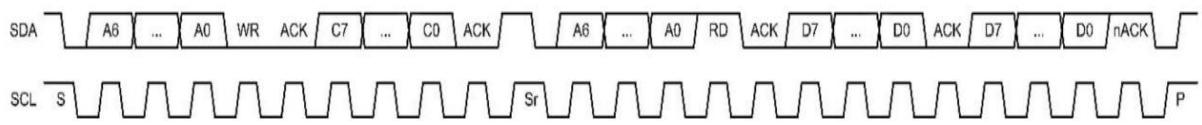
Start: IIC起始信号

器件地址(W): 器件地址 $0xB0+W(0)$ 表示写 (已转换为8位)

寄存器地址: 1字节, 见下表1具体描述

数据: 1字节或连续N字节($N \leq 16$), 见“具体寄存器描述”

读时序：



Start+器件地址(W)+寄存器址+Restart+器件地址(R)+回读的数据+Stop

解析：

Start: IIC起始信号

器件地址(W): 器件地址 $0xB0+W(0)$ 表示写 (已转换为8位)

寄存器地址: 1字节, 见下表1具体描述

Restart: 重发的Start信号

器件地址(R): 器件地址 $0xB0+R(1)$ 表示读 (已转换为8位)

数据: 回读的数据N个字节($N \leq 16$), 根据每个命令的数据量而定

Stop: IIC终止信号

通讯协议

寄存器表

地址	名称	R/W	描述	推荐值
00H	ID	R	IIC ID	0x58
01H	CTRL	R/W	控制寄存器	0x09
04H	TP HIGH BYTE	R	目标温度ADC数据高字节	0x00
05H	TP MIDDLE BYTE	R	目标温度ADC数据中字节	0x00
06H	TP LOW BYTE	R	目标温度ADC数据低字节	0x00
07H	TS HIGH BYTE	R	环境温度ADC数据高字节	0x00
08H	TS LOW BYTE	R	环境温度ADC数据低字节	0x00
0BH	START_REG	R/W	启动寄存器	0x80
23H	CFG_SYS	R/W	系统配置寄存器	0x20
24H	CFG_OPER	R/W	工作模式配置寄存器	0x15
25H	CFG_OSR	R/W	OSR配置寄存器	0x6D
26H	CFG_MEAS	R/W	测量配置寄存器	0x05
27H	CFG_DAC	R/W	DAC配置寄存器	0x00
28H	TS_BGR	R/W	环温TRIM寄存器	FUSE读取值
29H	OSC_TRIM	R/W	内部晶体TRIM寄存器	FUSE读取值
2AH	TP AGAIN	R/W	目标温度ADC放大增益	0xAA
2BH	TS AGAIN	R/W	环境温度ADC放大增益	0x14
34-35H	TS_Calib_Data_L	R	低环境温度校准值	0x00
4B-4CH	TS_Calib_Data_H	R	高环境温度校准值	0x00
3A-3DH	value_read_point.fValue	R	目温校准系数1	0x00
45-48H	value_read_bias.fValue	R	目温校准系数2	0x00
49H	ucFUSE_write_flag	R	目温系数校准标志	0xA5

通讯协议

简要操作流程

STEP1：环温目温校准系数读取

- 1: 读取34H开始的2个字节寄存器，获取TS_Calib_Data_L低温环温校准系数；
- 2: 读出4bH开始的2个字节，获取TS_Calib_Data_H高温环温校准系数；
- 3: 读取3AH开始的4个字节寄存器，获取目温校准系数value_read_point.fValue；
- 4: 读取45H开始的5个字节寄存器，获取目温校准系数value_read_bias.fValue和目温校准标志；

STEP2：配置工作模式

- 1: 按照推荐值配置23H-2BH寄存器；
- 2: 可以将配置完毕的值回读对比(建议操作)；

STEP3：启动器件工作

- 1: 配置01H寄存器为0x09设置连续工作模式；
- 2: 配置0BH寄存器为0x80启动转换；

STEP4：读目标ADC和环温ADC数据

- 1: 读取01H寄存器查看转换状态，如果BIT5为1代表本次转换完成，数据可用；
- 2: 读出04H开始的5个字节；分别按24BIT和16BIT拼接好放入内存；

STEP5：温度计算

- 1: 套用公式计算当前的环境温度，这里要借助STEP1读出的校准值；具体见下示例代码
- 套用器件的示例代码分驱动层和应用层。驱动层主要是IIC的底层操作，说明如何配置传感器，应用层的代码以测温为主，详见另一份代码移植说明。

以下就IIC底层驱动提供示例：

```
//变量定义
#define ADC_Max_Filter_Counter 7
Uint16_t VPATA_20_AD_Value; //整合的低温环温校准系数 对应TS_Calib_Data_L
Uint16_t VPATA_40_AD_Value; //整合的高温环温校准系数 对应TS_Calib_Data_H
Float PATA_Coe_K; //环温曲线K
Float PATA_Coe_B; //环温曲线B
Uint8_t data_buf[16]; //读取缓存
INT16U T_Sensor_Org_ADC; // TS瞬时值, 无符号
INT32U P_Sensor_Org_ADC; // TP瞬时值, 有符号
INT16U T_Sensor_ADC_Buf[ADC_Max_Filter_Counter];
INT32S P_Sensor_ADC_Buf[ADC_Max_Filter_Counter];
INT8U ADC_Max_Passed_Counter;
INT16U VPATA_AD_Value;
INT32S VObj_AD_Value;
Float PATA_Value; //环境温度值
Float Vobj_Temp_Real; //目标温度值
Uint8 _t ucFUSE_write_flag;

INT8U ucRead[10];
INT8U IIC_Redbuf[20];
INT8U IIC_Sendbuf[20];
typedef union
{
    float fValue;
    unsigned char ucArray[4];
}store_value_type;
```

通讯协议

```
store_value_type value_read_bias; //目温系数其一
store_value_type value_read_point; //目温系数其二

//基本函数
//环温系数计算方法 1 (适配环温范围较宽的应用)
//读取环温校准值(根据环温40度标定点计算环温, K值为固定值)
Void Read_TS_Calib_Data()
{
    IIC_Rec_Data_Process(0x4b, IIC_Recbuf,2);
    VPATA_40_AD_Value= IIC_Recbuf[0]*256+ IIC_Recbuf[1];//整合并拼凑校准系数
    PATA_Coe_K=55.2 ; //环温曲线K值
    PATA_Coe_B= VPATA_40_AD_Value -PATA_Coe_K*40.0;//计算环温曲线B值
}

//环温系数计算方法 2 (适配日常环温精准测温的应用)
//读取环温校准值(根据环温20度和环温40度两个标定点计算环温, K值为计算值)
Void Read_TS_Calib_Data()
{
    IIC_Rec_Data_Process(0x34, IIC_Recbuf,2);
    VPATA_20_AD_Value= IIC_Recbuf[0]*256+ IIC_Recbuf[1];//整合并拼凑校准系数1
    IIC_Rec_Data_Process(0x4b, IIC_Recbuf,2);
    VPATA_40_AD_Value= IIC_Recbuf[0]*256+ IIC_Recbuf[1];//整合并拼凑校准系数2
    PATA_Coe_K=(float)(VPATA_40_AD_Value-VPATA_20_AD_Value)/20.0;
    //计算K值
    PATA_Coe_B= VPATA_20_AD_Value -PATA_Coe_K*20.0 ;
    //计算环温曲线B值
}
```

备注：以上两种环温系数计算方法可根据需要选取其中一种。

```
//读取目温校准值和目温校准标志
Void Read_TP_Calib_Data()
{
    delay_100ms(1);
    IIC_Rec_Data_Process(0x3A,ucRead,4); //读取目温系数
    memcpy( (unsigned char *)(&value_read_point.ucArray[0]), (unsigned char *)
    )(&ucRead[0]), 4 );
    delay_100ms(1);
    memset(ucRead, 0x00, 10);
    IIC_Rec_Data_Process(0x45,ucRead,5); //读取目温系数和校准标志
    memcpy( (unsigned char *)(&value_read_bias.ucArray[0]), (unsigned char *)
    )(&ucRead[0]), 4 );
    ucFUSE_write_flag = ucRead[ 4 ];}
```

通讯协议

```

//写寄存器 含重读功能
int Temp_Write_Reg(INT8U address, INT8U *data_buf, INT8U len)
{
    static INT8U buf[128];
    int i;
    if( (len > 128) || (len < 2) )
        return -2;
    IIC_Send_Data_Process(address,data_buf,len);
    Delay_Long(2000);
    IIC_Rec_Data_Process(address, buf, len);
    for(i=0;i<len;i++) {
        if(data_buf[i] != buf[i])
            return -1;
    }
    return 0;
}

//初始化配置并启动器件工作
Void Ini_ADC_Chip(void)
{
    Uint8_t buf[]={0x20,0x15,0x6d,0x05,0x00,0x00,0x00,0xaa,0x14};
    IIC_Rec_Data_Process(0x28,ucRead,2);
    buf[5]=ucRead[0];
    buf[6]=ucRead[1];
//配置23H寄存器开始的9个单元，其中0x28、0x29寄存器使用的是从FUSE读取的数值，其它7个寄存器使用的是推荐值。
//IIC_Send_Data_Process(0x23,buf,9); //写Fuse寄存器 不含重读功能
while(Temp_Write_Reg(0x23,buf,9)); //写Fuse寄存器 含重读功能
delay_10us(2);
    IIC_Sendbuf[0]=0x09;
    IIC_Send_Data_Process(0x01,IIC_Sendbuf,1); // 配置01H寄存器为0x09
delay_10us(2);
IIC_Sendbuf[0]=0x80;
IIC_Send_Data_Process(0x0b,IIC_Sendbuf,1); //配置0bH寄存器为0x80，启动转换
}

//读取器件的ADC数据
uint8_t ADC_Data_Ready(void)
{
    IIC_Rec_Data_Process(0x01, IIC_Redbuf,1); //从01H寄存器读一个字节到IIC_Redbuf内
    if(IIC_Redbuf[0]&0x20) //查看数据是否Ready
    {
        delay_10us(2);
        IIC_Rec_Data_Process(0x04, IIC_Redbuf,5); //从04H寄存器连续读5个字节到
d      a          t      a          b      u      f      内
T_Sensor_Org_ADC=(INT16U)IIC_Redbuf[3]*0x100+(INT16U)IIC_Redbuf[4];
        //整合环境温度ADC数据 //整合目标温ADC数据
P_Sensor_Org_ADC=(INT32U)IIC_Redbuf[0]*0x10000+(INT32U)IIC_Redbuf[1]*0x100+
(INT32U)IIC_Redbuf[2];
        return 1 ;
    }
    return 0 ;
}

```

通讯协议

```
//排序函数
void Bubble_Array_16_High(INT16U *bArray, INT16U iFilterLen)
{
    INT16U i,j;// 循环变量
    INT16U bTemp;
    // 用冒泡法对数组进行排序
    for (j = 0; j < iFilterLen - 1; j++)
    {
        for (i = 0; i < iFilterLen - j - 1; i++)
        {
            if (bArray[i] > bArray[i + 1])
            {
                // 互换
                bTemp = bArray[i];
                bArray[i] = bArray[i + 1];
                bArray[i + 1] = bTemp;
            }
        }
    }
}

//排序函数
void Bubble_Array_32_High(INT32S *bArray, INT16U iFilterLen)
{
    INT16U i,j;// 循环变量
    INT32S bTemp;
    // 用冒泡法对数组进行排序
    for (j = 0; j < iFilterLen - 1; j++)
    {
        for (i = 0; i < iFilterLen - j - 1; i++)
        {
            if (bArray[i] > bArray[i + 1])
            {
                // 互换
                bTemp = bArray[i];
                bArray[i] = bArray[i + 1];
                bArray[i + 1] = bTemp;
            }
        }
    }
}

//ADC数据处理
INT8U ADC_Data_Sample_And_Filter(void)
{
    INT32U sum_ts=0;
    INT32S sum_tp=0;
    INT8U i;
```

通讯协议

```
if(ADC_Data_Ready())
{
    //P-sensor取15-16bit即可
    if(P_Sensor_Org_ADC&0x00800000)           // 符号位
    {
        P_Sensor_Org_ADC=P_Sensor_Org_ADC>>6;
        P_Sensor_Org_ADC|=0xffffc0000;

        P_Sensor_ADC_Buf[ADC_Max_Passed_Counter]=(INT32S)P_Sensor_Org_ADC;
    }
    else
    {
        P_Sensor_Org_ADC=P_Sensor_Org_ADC>>6;

        P_Sensor_ADC_Buf[ADC_Max_Passed_Counter]=(INT32S)P_Sensor_Org_ADC;
        } //T-sensor取所有的位数
        T_Sensor_ADC_Buf[ADC_Max_Passed_Counter]=T_Sensor_Org_ADC;
        ADC_Max_Passed_Counter++;

        if(ADC_Max_Passed_Counter>=ADC_Max_Filter_Counter)
        {
            ADC_Max_Passed_Counter=0;
            //排序

            Bubble_Array_16_High(T_Sensor_ADC_Buf,ADC_Max_Filter_Counter);

            Bubble_Array_32_High(P_Sensor_ADC_Buf,ADC_Max_Filter_Counter);

            sum_ts=T_Sensor_ADC_Buf[2]+T_Sensor_ADC_Buf[3]+T_Sensor_ADC_Buf[4];

            sum_tp=P_Sensor_ADC_Buf[2]+P_Sensor_ADC_Buf[3]+P_Sensor_ADC_Buf[4];
            VPATA_AD_Value=sum_ts/3;                      //滤过波的环境AD值
            VObj_AD_Value=sum_tp/3;                        //滤过波的目标AD值
            return 1;
        }
    }
    return 0;
}

//计算当前环境温度
//这个函数可以放在驱动端也可以放在应用端，因为计算出来的环境温度是FLOAT型，只要应用端节点调用便利，可以放在驱动端
void PATA_Temp_Caculate(void)
{
    float tempdata;
    tempdata=(float)(VPATA_AD_Value)-PATA_Coe_B;//利用之前计算出来的PATA_Coe_K和PATA_Coe_B变量
    PATA_Value=tempdata/PATA_Coe_K;//结合采集到的VPATA_AD_Value计算环境温度PATA_Value
}
```

通讯协议

```
//计算当前目标温度
void Object_Temprature_Caculate(void)
{
    float tempdata,tempdata1;
    if( ucFUSE_write_flag == 0xA5 )
    {
        tempdata = (float)VObj_AD_Value + value_read_bias.fValue;
        tempdata = tempdata / value_read_point.fValue;

        tempdata1=(PATA_Value+273.15)*(PATA_Value+273.15)*(PATA_Value+273.15)*(PATA_Value+273.15);
        tempdata=tempdata+tempdata1;
        Vobj_Temp_Real=pow(tempdata,1.0/4)-273.15; //最终温度
    }
}

}

//系统测温函数
void System_Temp_Caculate_Process(void)
{
    if(ADC_Data_Sample_And_Filter())//采样正常
    {
        PATA_Temp_Caculate(); //计算环境温度
        Object_Temprature_Caculate(); //计算目标温度
    }
}

//进入休眠模式
void Sensor_Sleep()
{
    IIC_Sendbuf[0]=0x00;
    IIC_Send_Data_Process(0x01,IIC_Sendbuf,1); //器件进入休眠
}

//退出休眠模式
void Sensor_Exit_Sleep()
{
    IIC_Sendbuf[0]=0x01;
    IIC_Send_Data_Process(0x01,IIC_Sendbuf,1); //器件退出休眠
    delay_100ms(2); //少量延时
    Ini_ADC_Chip(); //重新初始化器件并启动转换
}

//主函数
int main (void)
```

通讯协议

```
delay_100ms(2);
    __disable_irq();
    SystemInit();      //系统初始化
    Ini_ADC_Chip();   //初始化寄存器配置并启动器件工作
    Read_TS_Calib_Data();          //读取环温校准值
                                Read_TP_Calib_Data();          //读取目温校准值和目温校准标志
    __enable_irq();
    while(1)
    {
        System_Temp_Caculate_Process();
    }
    return 0;
}
```

样品检验报告

No.	外观	尺寸 (mm)		温度精度±2.5°C	
		A 4.5±0.1mm	B 3±0.1mm	35°C	50°C
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					