



敏思传感技术有限公司

MinSi (Henan) Sensing Technology Co., Ltd.

产品承认书

SPECIFICATION

客户名称 CUSTOMER	
产品名称 PRODUCTION	热电堆数字温度传感器
产品型号 MODEL	MS-TP-43CPDG
版本号 VERSION NO	A2.0

敏思传感技术有限公司

电话：400-8638-732

网址：http:// www.minsisensor.com

邮箱：minsi@ minsisensor.com



客户确认 CUSTOMER CONFIRMATION	审 核 CHECKED BY	编 制 PREPARED BY
	张毅	王洪波



声明

本说明书版权属**敏思传感技术有限公司**(以下称本公司)所有, 未经书面许可, 本说明书任何部分不得复制、翻译、存储于数据库或检索系统内, 也不可以电子、翻拍、录音等任何手段进行传播。

感谢您使用**敏思传感**的系列产品。为使您更好地使用本公司产品, 减少因使用不当造成的产品故障, 使用前请务必仔细阅读本说明书并按照所建议的使用方法进行使用。如果用户不依照本说明书使用或擅自去除、拆解、更换传感器内部组件, 本公司不承担由此造成的任何损失。

您所购买产品的颜色、款式及尺寸以实物为准。

本公司秉承科技进步的理念, 不断致力于产品改进和技术创新。因此, 本公司保留任何产品改进而不预先通知的权力。使用本说明书时, 请确认其属于有效版本。同时, 本公司鼓励使用者根据其使用情况, 探讨本产品更优化的使用方法。

请妥善保管本说明书, 以便在您日后需要时能及时查阅并获得帮助。

敏思传感技术有限公司



产品描述:

MS-TP-43CPDG是一款具有读出电路I²C的数字式热电堆传感器, 在测量物体温度时不需要直接接触。热电堆芯片基于MEMS工艺将上百对热电偶串联而成, 热电堆吸收从被测量物体发射的红外线能量, 利用塞贝克原理, 传感器输出对应的电压, 从而检测出目标的温度。该传感器可在-20℃~+85℃范围内使用, 测试温度范围为-20~+300℃, 有广阔的应用空间。I²C接口用于各种应用接口与本设备通信。



特点:

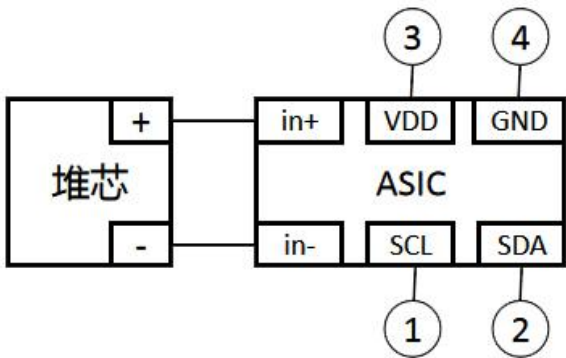
- ◆ TO-46 封装;
- ◆ 全集成数字红外热电堆;
- ◆ I²C 接口, 支持 PWM、继电器模式的输出;
- ◆ 稳定性好, 工作温度: -20℃~85℃;
- ◆ 内建高精度 24-Bit Sigma delta ADC, 以及校准 DSP 算法等;
- ◆ 高红外吸收层结构、高灵敏度;
- ◆ 测温范围: -20~+300℃;
- ◆ 测温精度: 100℃ 以上±2%, 人体范围区间±0.2℃。

主要应用:

- ◆ 非接触式温度测量;
- ◆ 体温, 额温等红外体温测量;
- ◆ 生产过程的连续温度控制;
- ◆ 人体存在式检测;
- ◆ 舒适性指数测量;
- ◆ 电力管理系统;
- ◆ 家用电器(空调、护发吹风机、油烟机等)温度测量与控制;
- ◆ 交互式电源控制;
- ◆ 照明单元控制。

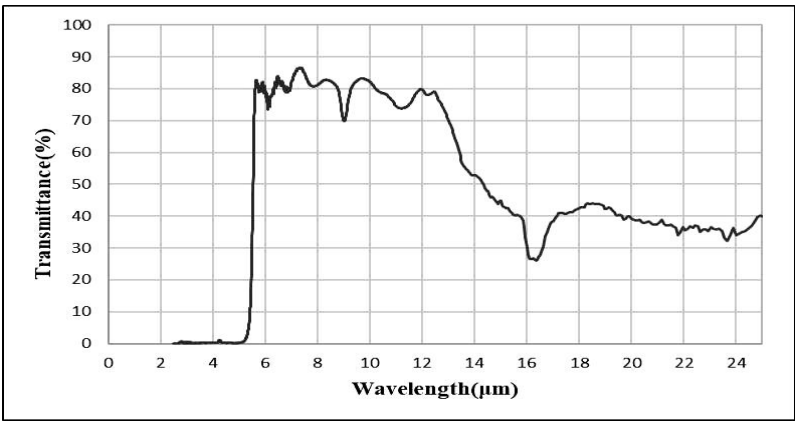
功能框图:

传感器被封装在具有 4 个引脚的密闭金属腔体中, 所用材料符合 RoHS 相关要求, 工作在 -20~85℃温度范围内。



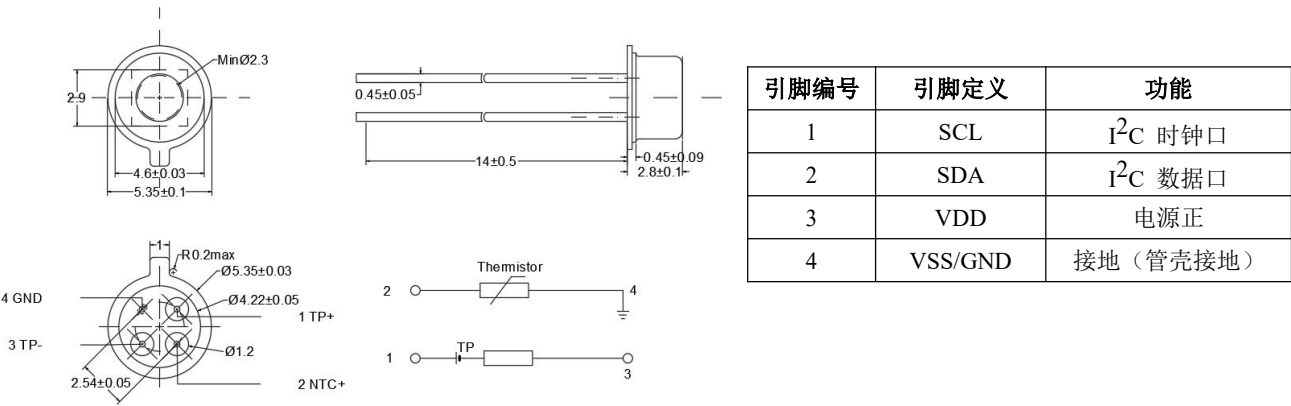
技术指标：

参数	单位	最小值	典型值	最大值
热电堆敏感面积	mm2	/	1.4×1.4	/
工作电压	V	2.3		3.6
工作电流	μA	/	507	800
睡眠电流	μA		5	
工作温度范围	℃	-20	/	+85
储存温度范围	℃	-40	/	+125
ESD性能 HMB	kV	/	±2kV	/
滤光片波长范围	μm	5.5	/	14
测温范围	℃	-70		+300
测温精度	℃	/	±1℃（大于 100 测温范围的±2%）	/

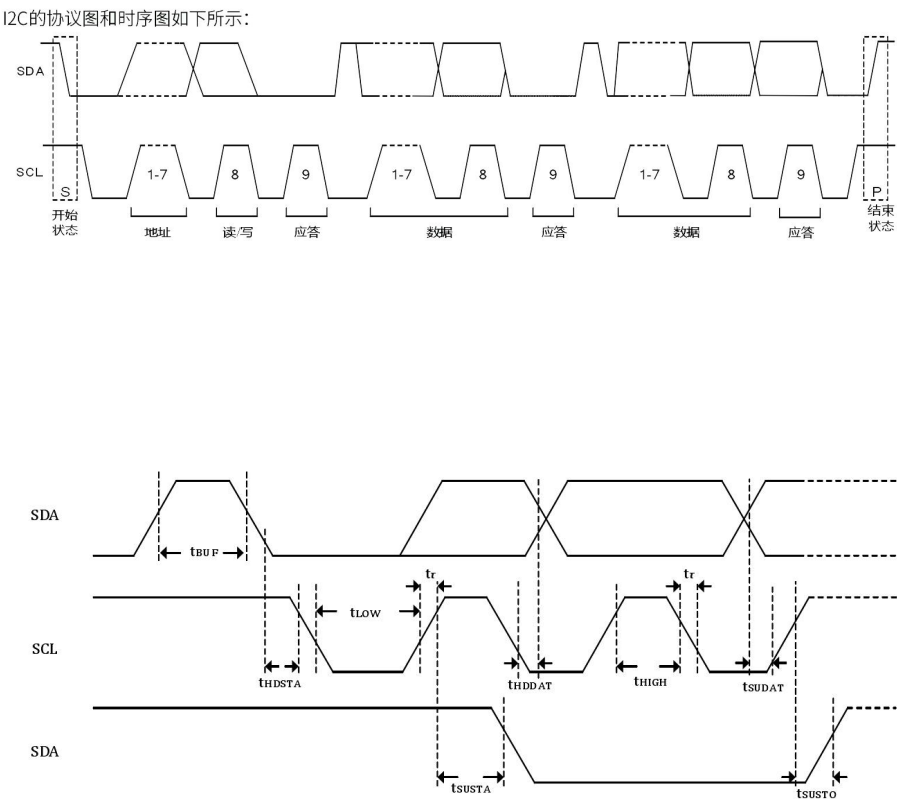




封装尺寸（单位：mm）



应用说明书：





标识	参数	最小值	最大值	单位
f _{BscIB}	时钟频率		400	kHz
t _{BLOWB}	时钟低脉冲维持时间	1.3		us
t _{BHIGHB}	时钟高脉冲维持时间	0.6		us
t _{BSUDATB}	SDA建立时间	0.1		us
t _{BHDDATB}	SDA保持时间	0.0		us
t _{BSUSTAB}	每次开始时的建立时间	0.6		us
t _{BHDSTAB}	开始条件保持时间	0.6		us
t _{BSUSTOB}	停止时间建立时间	0.6		us
t _{BBUF}	两次通讯之间间隔时间	1.3		us

地址	读写	数据位	寄存器名	默认值	描述
0x00	W	5,2	reset	0	软重启
0x02	R	3	Temp_rdy	0	Tobj 红外温度准备好
		0	Tamb_rdy	0	Tamb 本体温度准备好
0x10	R	7-0	MSB	0	Tobj 高 8 位: 最高位为符号位
0x11	R	7-0	CSB	0	Tobj 中 8 位
0x12	R	7-0	LSB	0	Tobj 低 8 位
0x16	R	7-0	MSB	0	Tamb 高 8 位: 最高位为符号位
0x17	R	7-0	CSB	0	Tamb 中 8 位
0x18	R	7-0	LSB	0	Tamb 低 8 位
0x30	RW	5	Sleep en	0	0:exit sleep 1:enter sleep
		4	clk mode	0	0: 600kHz 1: 1.2MHz
		3	Start	0	写入 1 启动探头温度转换运行
0x40	R	7	blow	0	写入 0x68 烧写 EEPROM
以下为 EEPROM 寄存器					
0x92		6-0	Chip address	0	I2C 地址, 0x7F 为默认地址 , 不管该寄存器为多少都可使用。 修改后需要烧写 EEPROM 才可使用。

温度计算方法 (以 C 语言为例):

```

int temp, reg[3];
float tempobj;
temp=0;
temp=reg[2]; // MSB 高 8 位
temp=(temp<<8)|reg[1]; // CSB 中 8 位
temp=(temp<<8)|reg[0]; // LSB 低 8 位
if (temp&0x00800000) temp|=0xFF000000; // 符号位判断
tempobj=(float)temp/16384; // 除以 2 的 14 次方得到温度值, 精度为小数点后
2 位

```



运行顺序:

- 1, 上电后在 0x30 寄存器写入 0x08;
- 2, 读 0x02 寄存器, 若数据为 0x09 则表明温度转换完成 (读取间隔超过 0.5s 可忽略)。
- 3, 读 0x10 到 0x12 寄存器根据温度计算方法得到物体温度。
- 4, 读 0x16 到 0x18 寄存器根据温度计算方法得到传感器本体温度。
- 5, 重复 2~4 步骤。

注意事项:

- 为减少传感器管脚之间的热干扰, 在制作 PCB 时, 应将传感器管脚之间进行热隔离;
- 手工焊接温度 330 ± 20 °C, 单个管脚焊接时间不超过 3s;
- 频繁、过度振动、强烈冲击或碰撞会导致传感器内部产生共振而断裂;
- 禁止用手或尖锐物品直接接触传感器窗口滤光片, 避免污染。

I²C 程序示例 (软件模拟, 请根据实际情况调整 IO 口配置: 设置为上拉开漏模式, 一次只能读取或写入一个字节, 不能连续读取写入)

unsigned char NSA_READ(unsigned char addr)

```
{
    unsigned char temp=0;
    IIC_Start();
    IIC_Send_Byte(NSA_ID_ADDR);
    IIC_Wait_Ack();
    IIC_Send_Byte(addr);
    IIC_Wait_Ack();
    IIC_Start();
    IIC_Send_Byte(NSA_ID_ADDR | 0x01);
    IIC_Wait_Ack();
    temp=IIC_Read_Byte(0);
    IIC_Stop();
    return temp;
}
```

void NSA_WRITE(unsigned char addr,unsigned char data)

```
{
    IIC_Start();
    IIC_Send_Byte(NSA_ID_ADDR);
    IIC_Wait_Ack();
```



```
IIC_Send_Byte(addr) ;
    IIC_Wait_Ack();
    IIC_Send_Byte(data) ;
    IIC_Wait_Ack();
    IIC_Stop();

}

void IIC_Start(void)
{
    SIIC_SDA_H;
    SIIC_SCL_H;
    delayus(10);
    SIIC_SDA_L; //START:when CLK is high,DATA change form high to low
    delayus(10);
    SIIC_SCL_L; //钳住 I2C 总线，准备发送或接收数据
    delayus(30);
}

void IIC_Stop(void)
{
    SIIC_SCL_L;
    SIIC_SDA_L;
    delayus(10);
    SIIC_SCL_H;
    delayus(10);
    SIIC_SDA_H; //发送 I2C 总线结束信号
    delayus(30);
}

//返回值： 1， 接收应答失败
//          0， 接收应答成功
u8 IIC_Wait_Ack(void)
{
    u8 ucErrTime=0;
    //delayus(10);
    SIIC_SDA_H;
    delayus(4);
    SIIC_SCL_H;
    delayus(8);
    while(SIIC_SDA)
    {
```



```
        delayus(1);
        ucErrTime++;
        if(ucErrTime>10)
        {
            IIC_Stop();
            return 1;
        }
    }
    SIIC_SCL_L;//时钟输出 0
    return 0;
}

void IIC_Ack(void)
{
    SIIC_SCL_L;
    SIIC_SDA_L;
    delayus(8);
    SIIC_SCL_H;
    delayus(8);
    SIIC_SCL_L;
}

void IIC_NAck(void)
{
    SIIC_SCL_L;
    SIIC_SDA_H;
    delayus(8);
    SIIC_SCL_H;
    delayus(8);
    SIIC_SCL_L;
}
//IIC 发送一个字节
//返回从机有无应答
//1, 有应答
//0, 无应答
void IIC_Send_Byte(unsigned char txd)
{
    unsigned char t;
    SIIC_SCL_L;//拉低时钟开始数据传输
    delayus(4);
    for(t=0;t<8;t++)
```



```

{
    //IIC_SDA=(txd&0x80)>>7;
    if((txd&0x80)>>7)    //
        SIIC_SDA_H;
    else
        SIIC_SDA_L;
    txd<<=1;
    delayus(4);    //对 TEA5767 这三个延时都是必须的
    SIIC_SCL_H;
    delayus(8);
    SIIC_SCL_L;
    delayus(4);
}
}
//读 1 个字节, ack=1 时, 发送 ACK, ack=0, 发送 nACK
u8 IIC_Read_Byte(unsigned char ack)
{
    unsigned char i, receive=0;
    //SDA 设置为输入
    for(i=0; i<8; i++)
    {
        SIIC_SCL_L;
        delayus(8);
        SIIC_SCL_H;
        receive<<=1;
        if(SIIC_SDA) receive++;
        delayus(8);
    }
    if (!ack)
        IIC_NAck(); //发送 nACK
    else
        IIC_Ack(); //发送 ACK
    return receive;
}

```