

温湿度传感器NSHT30使用设计指南

AN-12-0010

作者：Juanjuan Shao



温湿度传感器NSHT30使用设计指南

产品介绍

NSHT30是一款基于CMOS-MEMS的相对湿度（%RH）和温度（T）传感器。它在单芯片上集成了一个完整的传感器系统，包括电容式的相对湿度传感器、COMS温度传感器和信号处理器以及I²C数字通信接口，采用2.5mm×2.5mm×0.9mm的DFN空腔封装。其I²C接口的通信方式、极小的封装和低功耗特性使得NSHT30可以更广泛地集成到各种应用中。

本文档旨在通过提供操作指导、软件配置示例和硬件设计指南，让用户尽快熟悉NSHT30系列产品。

相关产品

- NSHT30-CLAR 2.5mm×2.5mm×0.9mm
- NSHT30-QDNR 2.5mm×2.5mm×0.9mm

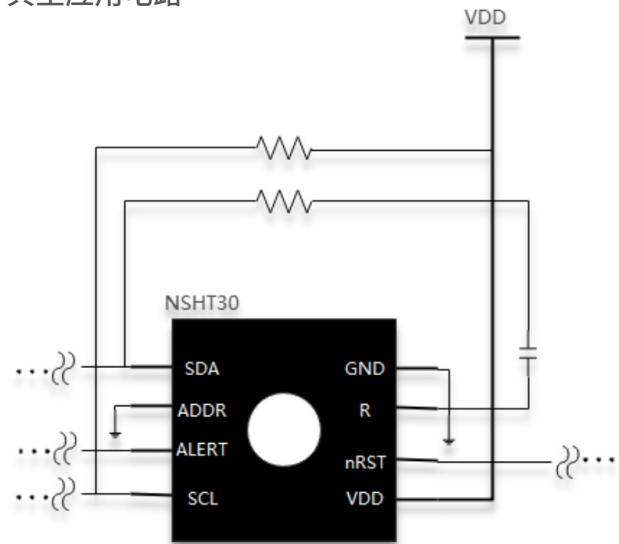
性能参数

- 相对湿度（%RH）传感器
 - 工作范围：0%RH~100% RH
 - 精度：±3% RH（典型值）
- 温度（T）传感器：
 - 工作温度范围：-40°C~125°C
 - 精度：±0.3°C（典型值）
- 相对湿度和温度补偿的数字输出
- 宽电源电压范围：2.0V~5.5V
- I²C数字接口，通信速度高达1 MHz
 - 两个可选地址
 - 带有CRC校验的数据保护
- 低功耗：平均电流3.2μA
- 8-Pin LGA 封装和DFN封装可选

应用

- ✓ 洗衣机&干衣机
- ✓ 智能恒温恒湿控制器
- ✓ 白电
- ✓ 打印机
- ✓ 加湿器/除湿器
- ✓ 医疗呼吸机
- ✓ 智能家居

典型应用电路



温湿度传感器NSHT30使用设计指南

目录

1. NSHT30 Devices	2
1.1. 采用LGA封装的NSHT30-CLAR	2
1.2. 采用DFN封装的NSHT30-QDNR	2
2. 芯片特性	3
2.1. 温湿度采集模式说明	3
2.1.1. 单次测量模式	3
2.1.2. 周期测量模式	4
2.2. 其它功能描述	5
2.2.1. Heater 功能	5
2.2.2. Reset功能	5
2.3. 芯片使用注意事项	6
2.3.1. 推荐使用环境	6
2.3.2. 避免接触相关化学物质	6
2.3.3. 极限条件下恢复方法	6
2.3.4. 长期漂移及评估加速模型	7
3. 储存和操作	7
3.1. 静电放电防护	7
3.2. 包装和储存	8
3.3. 焊接注意事项	8
3.4. 装配及使用中注意规避化学污染物	9
4. NSHT30设计参考	10
4.1. NSHT30硬件设计参考	10
4.1.1. DEMO原理图	10
4.1.2. Layout指导	12
4.2. NSHT30软件驱动范例	12
5. 传感器装配建议	20

温湿度传感器NSHT30使用设计指南

1.NSHT30 Devices

1.1.采用LGA封装的NSHT30-CLAR

NSHT30-CLAR采用金属壳（Metal Can Package）封装，封装中央开孔设计，对于相对湿度（%RH）传感器未做额外的保护，封装尺寸为2.5mm×2.5mm×0.9mm。使用NSHT30-CLAR时，应注意确保遵循所有要求。

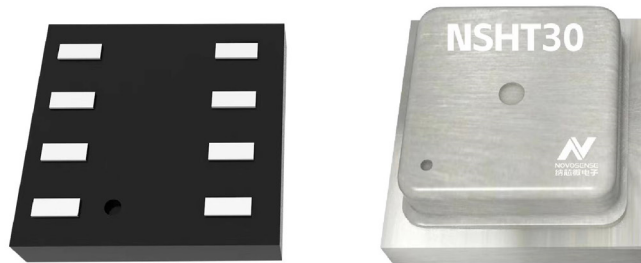


图1.1 采用Metal Can Package的NSHT30

1.2.采用DFN封装的NSHT30-QDNR

NSHT30-QDNR与NSHT30-CLAR封装和引脚兼容。该器件采用标准DFN Open Cavity封装，对于相对湿度（%RH）未做额外保护，封装尺寸为2.5mm×2.5mm×0.9mm。使用NSHT30-QDNR时，应注意确保遵循所有要求。



图1.2 采用DFN Open Cavity封装的NSHT30

温湿度传感器NSHT30使用设计指南

2. 芯片特性

2.1. 温湿度采集模式说明

2.1.1. 单次测量模式

NSHT30 有两种工作模式，其中Single shot mode 为单次测量模式，用户可发送hex code:2400 (high) /240B (Medium) /2416 (low) 触发该模式，不同指令对应不同的可重复性配置，详见下表。

表2.1 Single shot Mode 测量指令

Hex. code		Condition
MSB	LSB	Repeatability
24	00	High
	0B	Medium
	16	Low

芯片接受到单次转换命令后，根据指令对应的Repeatability配置，进行一次温湿度测量，并在测量完成后更新一次测量数据，测量数据包含温度值，温度CRC值，湿度值和湿度CRC值。用户可通过I²C总线，从芯片I²C地址回读6Bytes数据，根据CRC校验结果判定无异常后，根据转换公式转换为温度值，相对湿度值；具体可参考DataSheet及Demo Code部分。

温湿度传感器NSHT30使用设计指南

2.1.2.周期测量模式

Periodic mode为NSHT30 的另一种工作模式，即周期性测量模式。根据测量周期，可重复性的不同配置，周期性测量指令共有15种，其中mps为一秒钟进行测量的次数，表征测量周期。具体指令见表2.2。

表2.2 Periodic Mode 测量指令

Hex. code		Condition	
MSB	LSB	Periodic	Repeatability
20	32	0.5 mps	High
	24		Medium
	2F		Low
21	30	1 mps	High
	26		Medium
	2D		Low
22	36	2 mps	High
	20		Medium
	2B		Low
23	34	4 mps	High
	22		Medium
	29		Low
27	37	10 mps	High
	21		Medium
	2A		Low

芯片工作在周期性测量模式下，自动更新温湿度测量结果至最近一次完成测量的数据。温湿度测量数据的读取，需要用户发送数据读取指令（hex code:E0 00）至芯片后，从芯片回读温湿度数据。从芯片回读数据与单次测量模式相同。

在周期测量模式下的芯片，可通过向芯片发送退出指令（hex code:30 93）或者复位芯片退出。

温湿度传感器NSHT30使用设计指南

2.2.其它功能描述

2.2.1.Heater 功能

NSHT30支持Heater开启功能，开启后芯片的输出温度值变大几摄氏度，相对湿度输出值随之变小。Heater指令见表2.3。

表2.3 Heater 指令

Hex. code		Command
MSB	LSB	
30	60	Heater on
	66	Heater off

Heater功能可应用到以下场景：

- 1.Heater 开启后温度值和相对湿度值有改变，能够判断传感器是否处在正常工作状态。
- 2.测量环境出现冷凝现象，传感器表面出现冷凝，可通过开启Heater，加速传感器发生冷凝后的恢复；

2.2.2.Reset功能

NSHT30有三种复位方式，可以通过发送指令、通过nReset引脚或者通过对芯片下电进行复位。

表2.4 soft reset 指令

Hex. code		Command
MSB	LSB	
30	A2	Soft reset

另，不建议客户通过I²C General Call对IIC总线上芯片进行统一复位；NSHT30对I²C General Call复位指令会进行响应，但时序上存在特殊性，如要求使用General Call指令，请与我司联系获取相关支持。

温湿度传感器NSHT30使用设计指南

2.3. 芯片使用注意事项

2.3.1. 推荐使用环境

NSHT30的推荐工作环境应控制在0-80°C的温度范围内，相对湿度应控制在20-80%RH的范围内，在这个范围内能够保证传感器的性能达到最佳。

2.3.2. 避免接触相关化学物质

温湿度传感器是高精确的环境传感器，容易受到污染物的影响，必须防止暴露于易挥发性化学品、酸、碱和清洁剂。酮类、丙酮、乙醇、异丙醇、甲苯等可能导致传感器湿度精度产生不可逆漂移，在某些情况下甚至是不可逆的漂移。为了确保传感器湿度精度特性，必须遵守这些准则：

1. 避免将传感器暴露在挥发性化学品，如溶剂或有机化合物中。这些化学品通常存在于环氧树脂、胶水、粘合剂和塑料中，并可能在烘烤和固化过程中可能会放出气体，也会对传感器产生潜在的影响。
2. 避免接触强酸和强碱。
3. 避免使用清洁剂，如洗板水，高浓度酒精等。

2.3.3. 极限条件下恢复方法

如果传感器长时间暴露在推荐的操作环境之外（超过60小时），特别是当相对湿度小于3%RH或大于80%RH时，相对湿度的精度会发生暂时偏移。环境恢复正常后，偏移可恢复，但长期暴露在极端条件下可能会加速温湿度传感器的老化。传感器自恢复的过程相对缓慢。可通过以下方式使传感器加速恢复到正常状态：

- 烘烤：95°C，<5 %RH，12小时。
- 水合：25°C，90%RH，12小时。
- 室温放置12小时。

温湿度传感器NSHT30使用设计指南

2.3.4. 长期漂移及评估加速模型

NSHT30 long term drift折算模型遵循佩克模型。

$$AF = AF_H * AF_T = \left(\frac{H_A}{H_U}\right)^n * e^{\frac{E_a}{K} \left(\frac{1}{T_U} - \frac{1}{T_A}\right)}$$

其中AF为加速条件相对于使用环境的加速因子；AF_H为湿度加速因子，AF_T为温度加速因子；

n为模型参数，推荐取值范围1—5，是由腐蚀特性决定；K为玻尔兹曼常数；E_a为化学反应活化能，通常取值为0.65 eV；

H_A为加速试验相对湿度；H_U为使用环境相对湿度；

T_A为加速试验绝对温度；T_U为使用环境绝对温度；

NSHT30 long term drift折算依据HTOL的结果。

3. 储存和操作

3.1. 静电放电防护

为确保传感器的正常工作，必须在任何时候都要防止静电放电（ESD）。所有对传感器的处理都应专门在静电放电保护区（EPA）内进行，确保该保护区已被正确设置，以最大限度地减少ESD的风险。其中包括用腕带或类似的措施将人员接地，将所有的导电材料接地。

此外，所有操作都应在接地的导电地板上进行。为了进一步保护传感器，当不在EPA内处理时，应使用ESD保护材料对其进行包装。

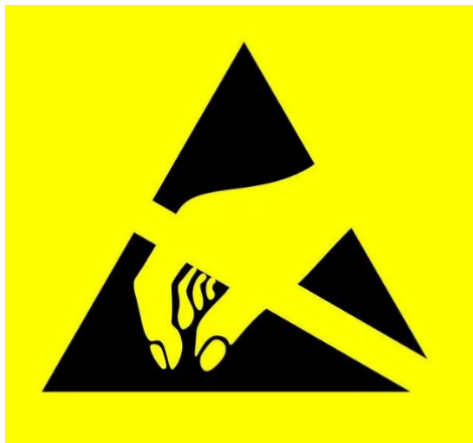


图3.1 静电敏感器件

温湿度传感器NSHT30使用设计指南

3.2. 包装和储存

纳芯微温湿度传感器采用密封性防静电卷带进行包装和储存。去除卷带之后，在使用之前可以将传感器存储在湿度和温度受控的环境中。贮存温度和湿度的限值取决于传感器的潮湿敏感度等级（MSL），其中NSHT30-CLAR遵循MSL3，NSHT30-DQNR遵循MSL1。

请勿将温湿度传感器储存在防静电聚乙烯袋或包装材料（粉色泡沫/包装）中，因为这些材料会散发可能对传感器产生影响的气体。NOVOSENSE建议使用防静电、可密封的袋子进行储存。请勿在储存容器内使用粘合剂或胶带。

3.3. 焊接注意事项

温湿度传感器可以使用标准的回流焊炉进行焊接。它能够承受焊接过程，包括符合IPC/JEDEC J-STD-020的曲线，峰值温度为260°C在红外/对流回流焊炉中进行无铅焊接时，可承受长达30秒的时间（见图3.2）。根据IPC/JEDEC J-STD-020标准，分类温度是根据封装厚度和封装尺寸来规定的。IPC/JEDEC J-STD-020标准中的无铅工艺分类温度（Tc）表列出了这些温度，见表3.1。

表3.1 无铅工艺分类温度(Tc)表

Package Thickness	Volume < 350mm ³	Volume < 350-2000mm ³	Volume < 2000mm ³
< 1.6mm	260°C	260°C	260°C
1.6mm-2.5mm	260°C	250°C	245°C
> 2.5mm	250°C	245°C	245°C

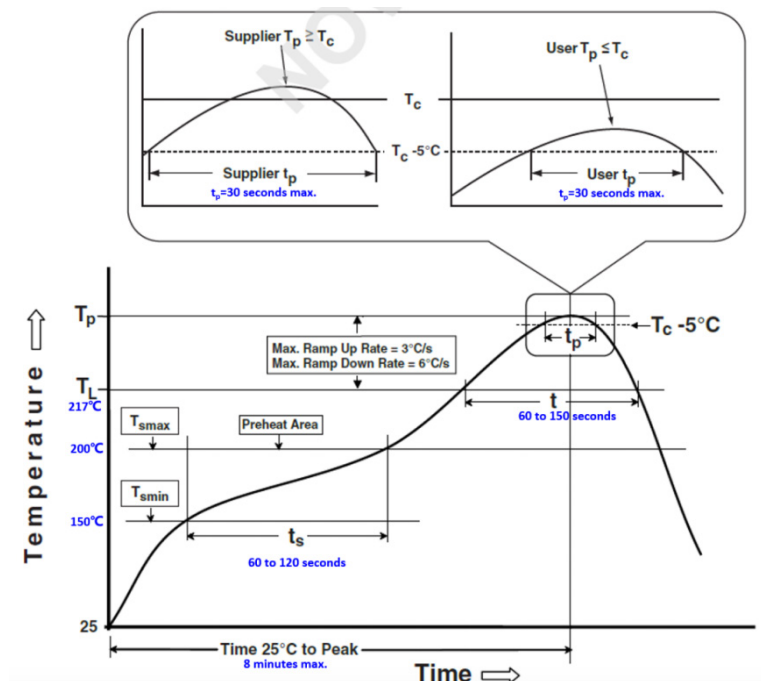


图3.2 回流焊炉温曲线

温湿度传感器NSHT30使用设计指南

图3.2阐述了回流焊各个温区的关键温度/时间关系。回流焊曲线的峰值温度 (T_p)，是在芯片封装顶部测量。

-对于IC供应商： T_p 必须等于或大于 T_c ，如表4.1所示。

-对于用户来说： T_p 必须不超过表4.1所示的 T_c 。

-定义峰值温度的时间(t_p)开始/结束于 $T_c-5^\circ\text{C}$ 。

在焊接温湿度传感器时，必须确保遵守最高温度和暴露时间，以防止损坏传感器。如果PCB经过多次焊接循环，我们建议在最后一个焊接周期进行温湿度传感器的焊接，以减少传感器损坏风险。强烈建议使用 "免清洗" 锡粉粒度T3以上的焊锡膏，因为它不需要洗板，而洗板可能会对传感器有害。

需要注意的是，温湿度传感器过回流焊应该少于三次，超过三次可能会造成焊接接触不良的问题。

因为NSHT30为开孔结构，Reflow后进行焊接前可以用高温胶带盖住芯片的孔洞，防止助焊剂等进入孔内，影响湿度精度的测量。

回流焊会使芯片暴露于极端环境中（高温低湿度），经过回流焊后，需要对芯片进行水合处理，处理方法参考2.3.3进行处理。

3.4. 装配及使用中注意规避化学污染物

温湿度传感器容易受到污染物的影响，必须防止暴露在易挥发性化学品、有机化合物、酸、碱、清洁剂、酮类、乙醇、异丙醇、甲苯、粘合剂等环境中，避免对传感器湿度精度产生不可逆的影响。

温湿度传感器NSHT30使用设计指南

4.NSHT30设计参考

4.1.NSHT30硬件设计参考

4.1.1.DEMO原理图

该Demo 的硬件包含两部分，一部分是基于51单片机的采集通讯模块，另一部分是焊接的NSHT30模块，通过排线连接两个模块，电路图及Layout见下图

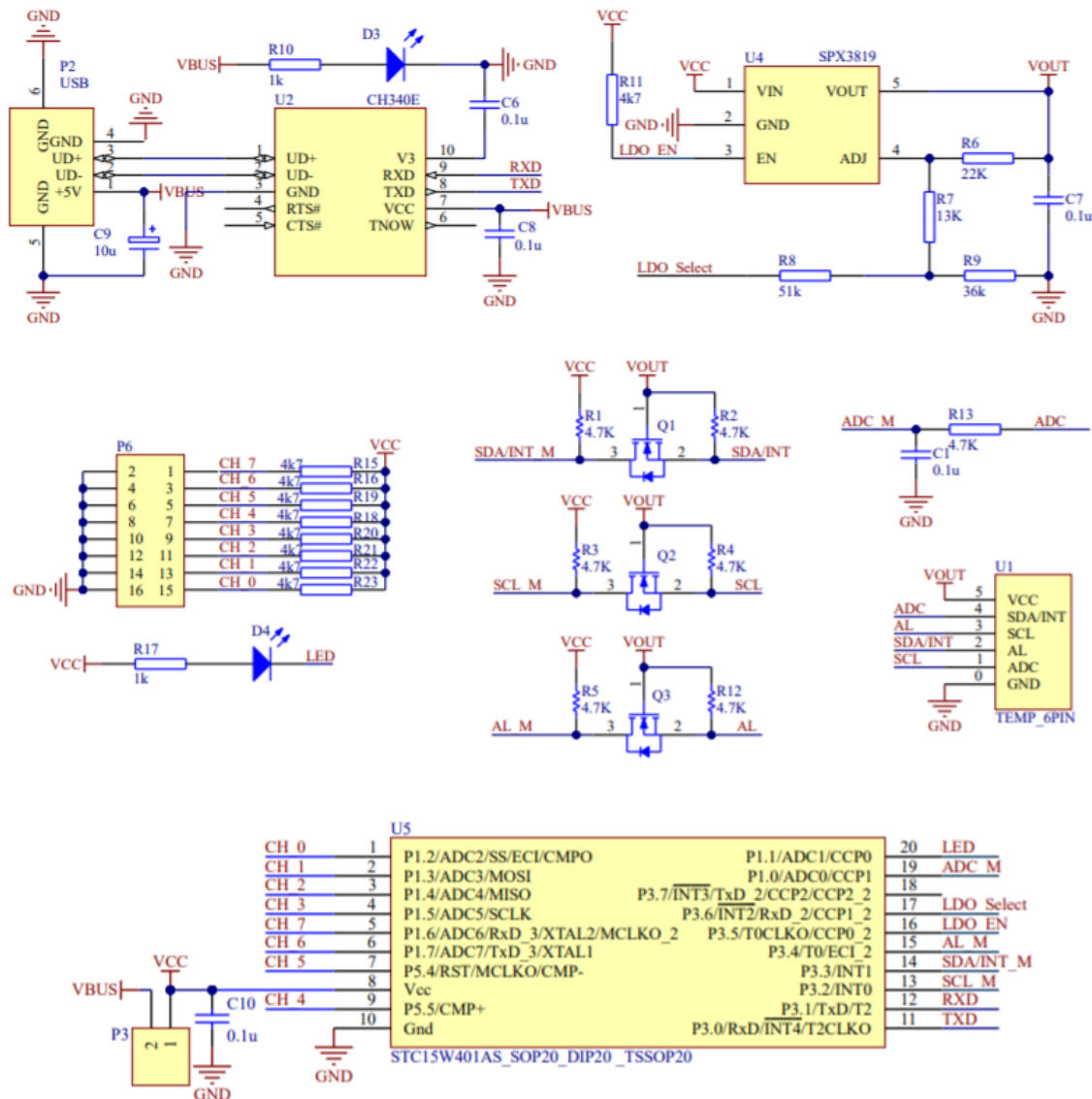


图4.1 采集通讯原理图

温湿度传感器NSHT30使用设计指南

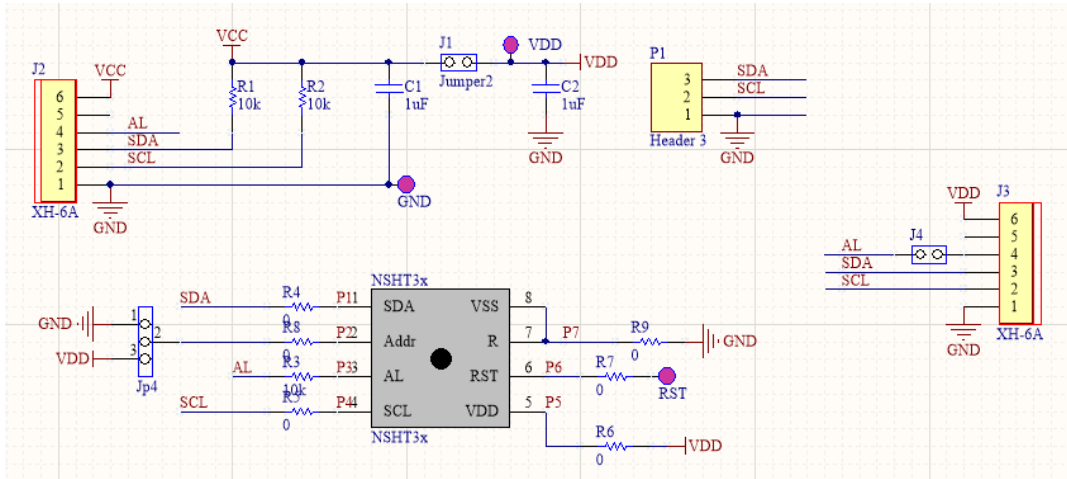


图4.2 NSHT30 应用原理图

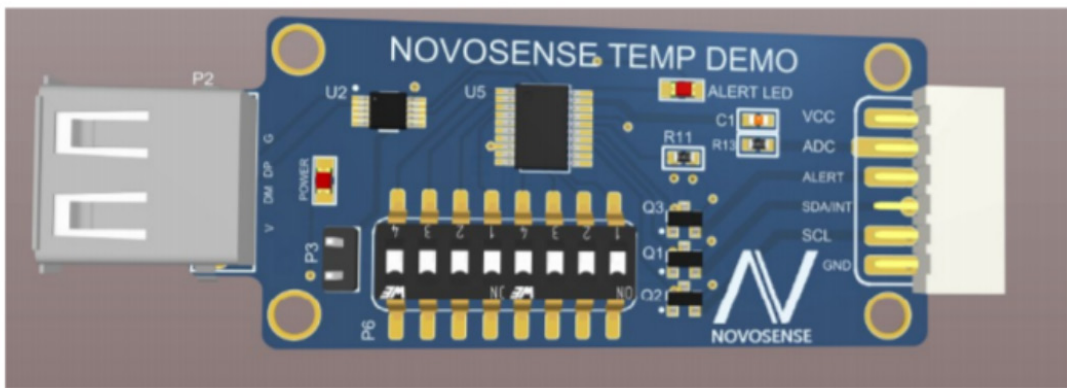


图4.3 基于 51 单片机的采集通信模块

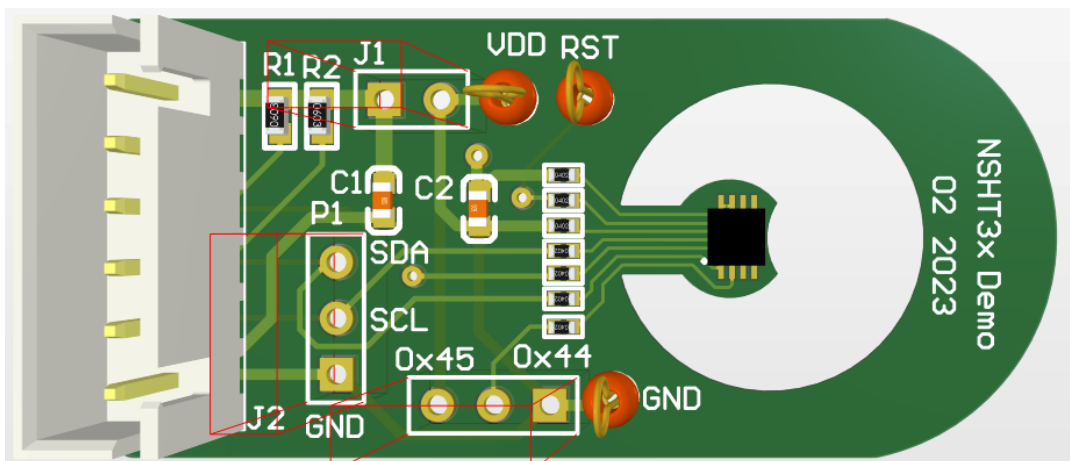


图4.4 NSHT30- Soldering PCB

温湿度传感器NSHT30使用设计指南

4.1.2.Layout指导

具体到PCB 的Layout建议:

- 1.温湿度传感器芯片位置尽量远离热源;
- 2.降低温湿度传感器芯片与PCB间的热传导;(芯片周围不覆铜,走线尽量细,PCB上芯片周围开孔,降低热传导,或者采用柔性软板将温湿度传感器与测试主板进行物理隔离)
- 3.降低温湿度传感器芯片耦合到PCB上之后的热惯性;(温湿度传感器所在的PCB尽量薄,例如采用单独的FPC软板装配温湿度传感器)

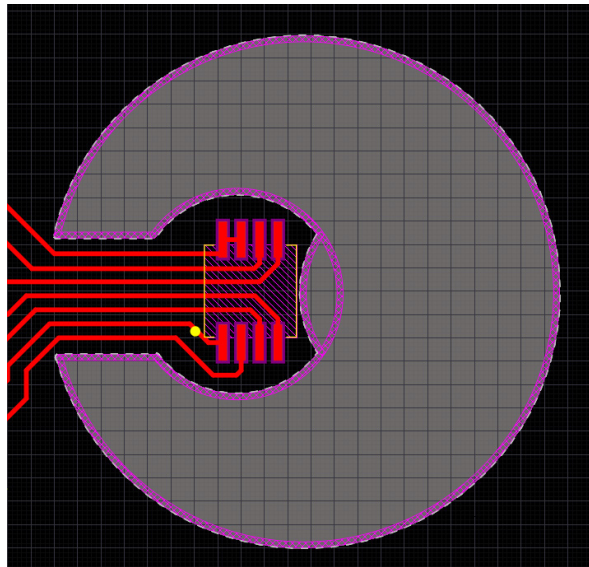


图4.5 布局示例

4.2.NSHT30软件驱动范例

```
*****
// IIC驱动代码
void IIC_Start(void)
{
    SDA_OUT();
    IIC_SDA=1;
    IIC_SCL=1;
    delay_us(1);
    IIC_SDA=0;//START:when CLK is high,DATA change form high to low
    delay_us(1);
}
```

温湿度传感器NSHT30使用设计指南

```
        IIC_SCL=0;
    }
void IIC_Stop(void)
{
    SDA_OUT();
    IIC_SCL=0;
    IIC_SDA=0;//STOP:when CLK is high DATA change form low to high
    delay_us(1);
    IIC_SCL=1;
    delay_us(1);
    IIC_SDA=1;
    delay_us(1);
}
uint8_t IIC_Wait_Ack(void)
{
    uint32_t ucErrTime=0;
    SDA_IN();
    IIC_SDA=1;delay_us(1);
    IIC_SCL=1;delay_us(1);
    while(READ_SDA)
    {
        ucErrTime++;
        if(ucErrTime>1000)
        {
            IIC_Stop();
            return 1;
        }
    }
    IIC_SCL=0;
    delay_us(2);
    return 0;
}

void IIC_Ack(void)
{
```

温湿度传感器NSHT30使用设计指南

```
        SDA_OUT();
        IIC_SCL=0;
        delay_us(2);
        IIC_SDA=0;
        delay_us(2);
        IIC_SCL=1;
        delay_us(2);
        IIC_SCL=0;
        delay_us(2);
        IIC_SDA=1;
    }
void IIC_NAck(void)
{
    SDA_OUT();
    IIC_SCL=0;
    delay_us(2);
    IIC_SDA=1;
    delay_us(2);
    IIC_SCL=1;
    delay_us(2);
    IIC_SCL=0;
    delay_us(2);
}
void IIC_Send_Byte(uint8_t txd)
{
    uint8_t t;
    SDA_OUT();
    IIC_SCL=0;
    for(t=0;t<8;t++)
    {
        IIC_SDA=(txd&0x80)>>7;
        txd<<=1;
        delay_us(2);
        IIC_SCL=1;
        delay_us(2);
        IIC_SCL=0;
```


温湿度传感器NSHT30使用设计指南

```

        delay_us(2);
    }
}
uint8_t IIC_Read_Byte()
{
    unsigned char i, receive=0;
    SDA_IN();
    for(i=0; i<8; i++)
    {
        IIC_SCL=0;
        delay_us(2);
        IIC_SCL=1;
        receive<<=1;
        if(READ_SDA)
            receive++;
        delay_us(2);
    }
    IIC_SCL=0;
    return receive;
}
//NSHT30驱动代码
#define CMD_MEAS_SINGLE_H 0x2400 // measurement: SINGLE Mode high repeatability
#define CMD_MEAS_SINGLE_M 0x240B // measurement: SINGLE Mode medium repeatability
#define CMD_MEAS_SINGLE_L 0x2416 // measurement: SINGLE Mode low repeatability
#define CMD_MEAS_PERI_05_H 0x2032 // measurement: periodic Mode 0.5 mps high repeatability
#define CMD_MEAS_PERI_05_M 0x2024 // measurement: periodic Mode 0.5 mps medium repeatability
#define CMD_MEAS_PERI_05_L 0x202F // measurement: periodic Mode 0.5 mps low repeatability
#define CMD_MEAS_PERI_1_H 0x2130 // measurement: periodic Mode 1 mps high repeatability
#define CMD_MEAS_PERI_1_M 0x2126 // measurement: periodic Mode 1 mps medium repeatability
#define CMD_MEAS_PERI_1_L 0x212D // measurement: periodic Mode 1 mps low repeatability
#define CMD_MEAS_PERI_2_H 0x2236 // measurement: periodic Mode 2 mps high repeatability
#define CMD_MEAS_PERI_2_M 0x2220 // measurement: periodic Mode 2 mps medium repeatability
#define CMD_MEAS_PERI_2_L 0x222B // measurement: periodic Mode 2 mps low repeatability
#define CMD_MEAS_PERI_4_H 0x2334 // measurement: periodic Mode 4 mps high repeatability
#define CMD_MEAS_PERI_4_M 0x2322 // measurement: periodic Mode 4 mps medium repeatability

```

温湿度传感器NSHT30使用设计指南

```
#define CMD_MEAS_PERI_4_L 0x2329 // measurement: periodic Mode 4 mps low repeatability
#define CMD_MEAS_PERI_10_H 0x2737 // measurement: periodic Mode 10 mps high repeatability
#define CMD_MEAS_PERI_10_M 0x2721 // measurement: periodic Mode 10 mps medium repeatability
#define CMD_MEAS_PERI_10_L 0x272A // measurement: periodic Mode 10 mps low repeatability
void nsht30_set_periodic(uint8_t addr, uint16_t cmd)
{
    IIC_Start();
    IIC_Send_Byte(addr<<1);
    IIC_Wait_Ack();
    IIC_Send_Byte((cmd & 0xFF00) >> 8);
    IIC_Wait_Ack();
    IIC_Send_Byte(cmd & 0xFF);
    IIC_Wait_Ack();
    IIC_Stop();
}
//periodic mode
void nsht30_read_raw_periodic(uint8_t addr, uint8_t *buff)
{
    IIC_Start();
    IIC_Send_Byte(addr<<1);
    IIC_Wait_Ack();
    IIC_Send_Byte(0xE0); //Read cmd 0xE000
    IIC_Wait_Ack();
    IIC_Send_Byte(0x00);
    IIC_Wait_Ack();
    delay_ms(10);
    IIC_Start();
    IIC_Send_Byte((addr<<1)+1);
    if(IIC_Wait_Ack()==0)
    {
        buff[0]=IIC_Read_Byte();
        IIC_Ack();
        buff[1]=IIC_Read_Byte();
        IIC_Ack();
        buff[2]=IIC_Read_Byte();
    }
}
```

温湿度传感器NSHT30使用设计指南

```

        IIC_Ack();
        buff[3]=IIC_Read_Byte();
        IIC_Ack();
        buff[4]=IIC_Read_Byte();
        IIC_Ack();
        buff[5]=IIC_Read_Byte();
        IIC_NAck();
        IIC_Stop();
    }
}
//Single mode
void nsht30_read_raw_single(uint8_t addr, uint16_t cmd, uint8_t *buff)
{
    IIC_Start();
    IIC_Send_Byte(addr<<1);
    IIC_Wait_Ack();
    IIC_Send_Byte((cmd & 0xFF00) >> 8);
    IIC_Wait_Ack();
    IIC_Send_Byte(cmd & 0xFF);
    IIC_Wait_Ack();
    IIC_Stop();
    delay_ms(10);
    IIC_Start();
    IIC_Send_Byte((addr<<1)+1);
    if(IIC_Wait_Ack()==0)
    {
        buff[0]=IIC_Read_Byte();
        IIC_Ack();
        buff[1]=IIC_Read_Byte();
        IIC_Ack();
        buff[2]=IIC_Read_Byte();
        IIC_Ack();
        buff[3]=IIC_Read_Byte();
        IIC_Ack();
        buff[4]=IIC_Read_Byte();
        IIC_Ack();
    }
}

```

温湿度传感器NSHT30使用设计指南

```

        buff[5]=IIC_Read_Byte();
        IIC_NAck();

        IIC_Stop();
    }
}
//CRC calculation
int nst3x_crc_check(uint8_t *data, uint8_t len, uint8_t checksum)
{
    uint8_t crc = 0xFF, bit;
    uint8_t byteCtr;
    //calculates 8-Bit checksum with given polynomial
    for (byteCtr = 0; byteCtr < len; ++byteCtr)
    {
        crc ^= (data[byteCtr]);
        for (bit = 8; bit > 0; --bit)
        {
            if (crc & 0x80)
                crc = (crc << 1) ^ 0x131;
            else
                crc = (crc << 1);
        }
    }
    if (crc == checksum)
        return 1;
    else
        return 0;
}
//Reading and calculating temperature and humidity
HAL_StatusTypeDef read_temp_rh_1ch(uint8_t addr, double *pout)
{
    uint8_t dat[6];
    uint16_t tem,hum;
    nsht30_read_raw_single(addr,CMD_MEAS_SINGLE_L,dat);// single mode
    tem = ((uint16_t)dat[0]<<8) | dat[1];
    hum = ((uint16_t)dat[3]<<8) | dat[4];
}

```

温湿度传感器NSHT30使用设计指南

```

if((nst3x_crc_check(dat,2,dat[2])) && (nst3x_crc_check(dat+3,2,dat[5])))
{
    pout[0]= (175.0*(double)tem/65535.0-45.0) ;// T = -45 + 175 * tem / (2^16-1)
    pout[1]= (100.0*(double)hum/65535.0);// RH = hum*100 / (2^16-1)
    return HAL_OK;
}
else
{
    return HAL_ERROR;
}
}
//主函数
double rth[2];
int main(void)
{
    uint8_t addr = 0x44;
    MX_GPIO_Init();
    MX_USART2_UART_Init();
    IIC_Init();
    while (1)// 1s进行1次循环测量
    {
        delay_ms(1000);
        if(read_temp_rh_1ch(addr,rth) == HAL_OK)
        {
            printf("%3.4f,%3.6f%%\r\n",rth[0],rth[1]);
        }
        else
        {
            printf("crc error\r\n");
        }
    }
}

```

NSHT30部分驱动例程见上述代码，相关功能采用相应指令触发即可，无需预先进行Part ID识别，如需了解更多代码，请与我司联系获取相关支持。

温湿度传感器NSHT30使用设计指南

5. 传感器装配建议

◆ 为增加传感器与测试环境接触，建议装配时遵循几个基本原则：

1. 温湿度传感器与测试环境间必须具备物理接触通道；

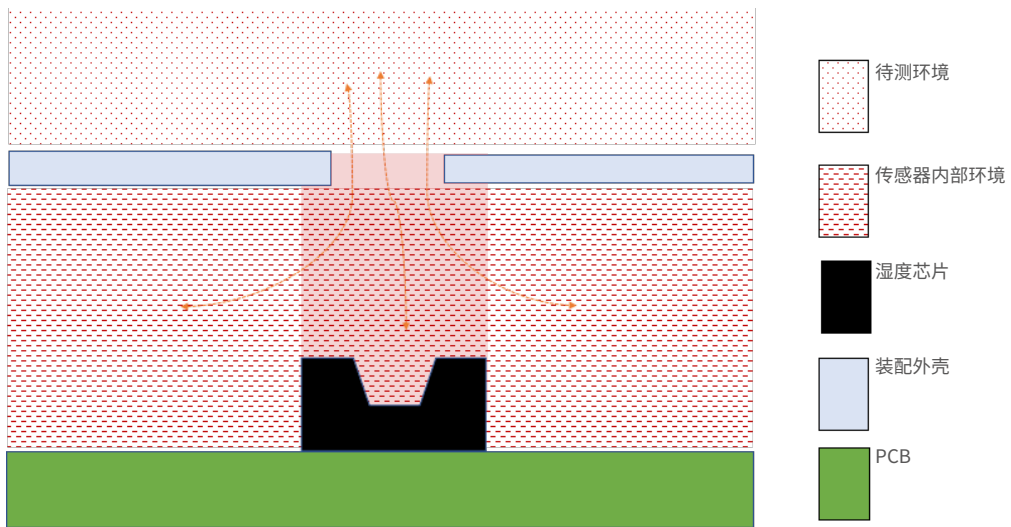


图5.1 芯片与环境良好接触

2. 将传感器区域跟壳体内部进行物理隔离，以提高响应速度，降低壳体内部空气对相对湿度测试值的影响；

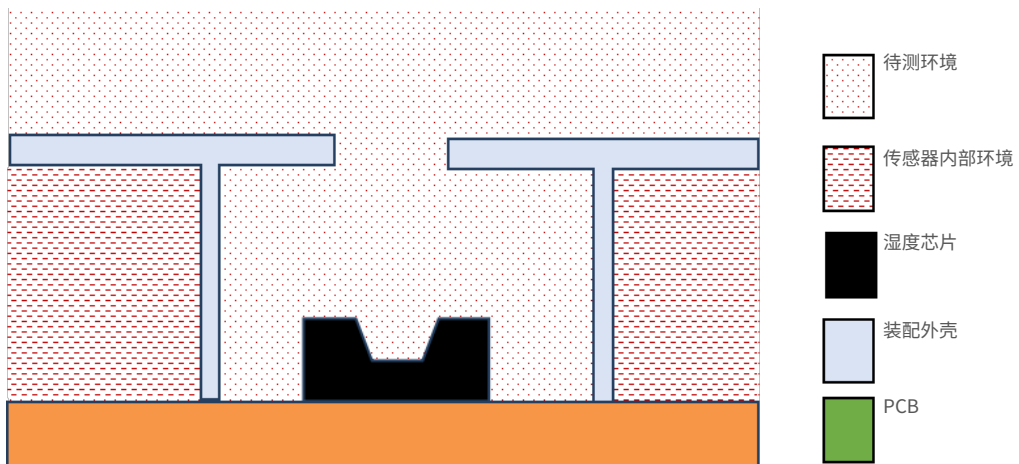


图5.2 芯片与壳内物理隔离

温湿度传感器NSHT30使用设计指南

3. 缩小传感器与壳体之间形成的区域体积，进一步提高对待测环境的响应时间；

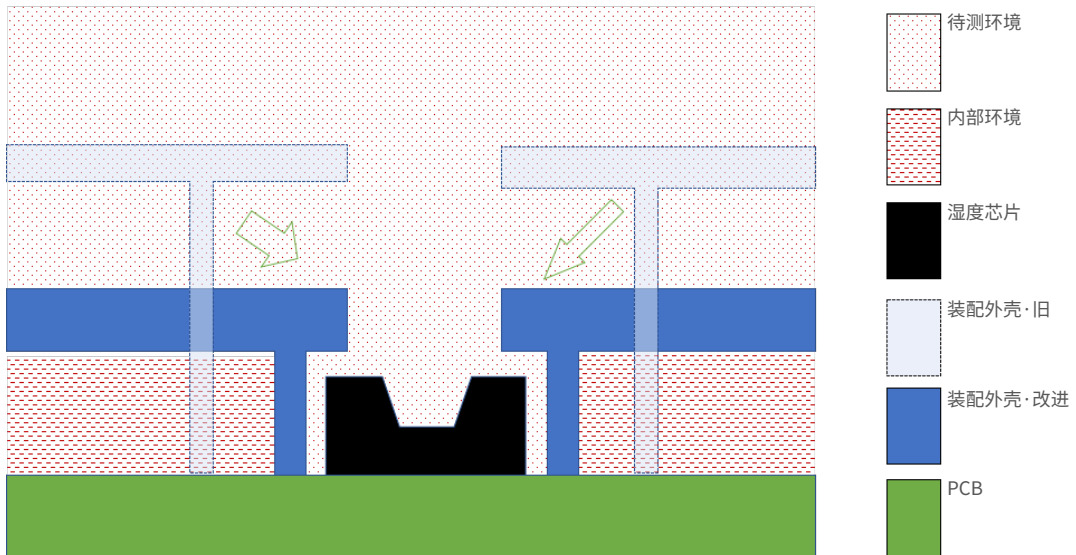


图5.3 减少芯片与壳体的区域体积

4. 如使用环境恶劣，有粉尘或者水溅射的风险，可在壳体开口处增加防水透气膜，防止传感器污染或者进水；

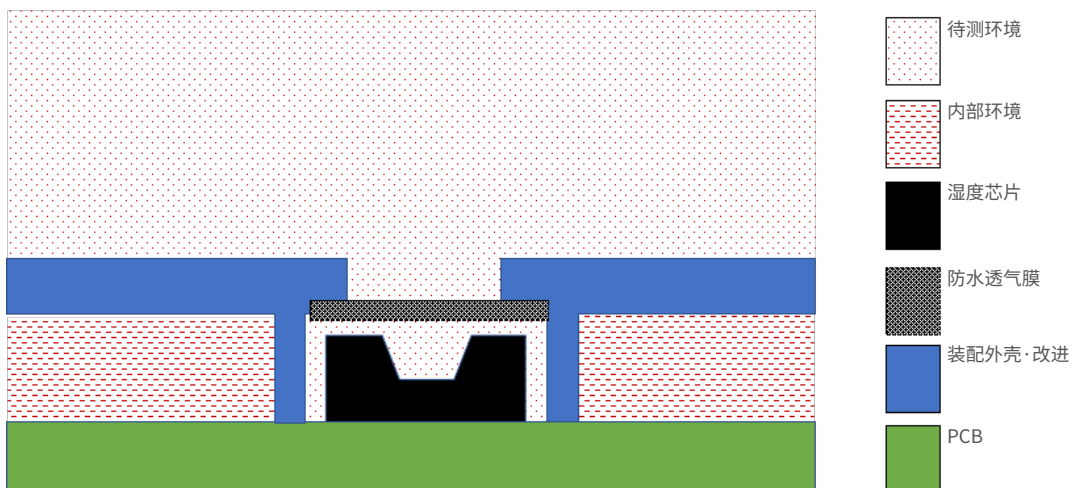


图5.4 增加防水透气膜

温湿度传感器NSHT30使用设计指南

- ◆ 从相对湿度基本定义出发，几个基本原则如下；

相对湿度定义

$$\text{相对湿度} = \frac{\text{当前空气中水蒸气含量}}{\text{当前空气能吸收的最大水蒸气含量}}$$

传感器与待测环境充分接触，空气中水蒸气的含量在传感器区域与待测环境间可基本保持一致；但是当前空气能够吸收的最大水蒸气浓度，与空气温度相关，温度越高，空气能够吸收的最大水蒸气浓度越高；因此，只有传感器测量温度与待测环境温度一致性很好的前提下，测量到的相对湿度值才是想要监控区域的相对湿度真实值；

以保证传感器温度与待测环境温度一致为出发点；需要从如下三个方向进行改善：

1. 传感器所在位置尽量接近需要测量的区域；测量区域与传感器间避开发热源，以免待测区域空气经发热源加热导致传感器区域相对湿度与测量区域相对湿度不一致；
2. 降低传感器区域的热惯性；提高对空气温度变化的跟随能力；几个基本方法如下：
 - 1) 降低PCB厚度；中间热焊盘尽量不焊接或者焊接焊盘与PCB覆铜隔离开来；

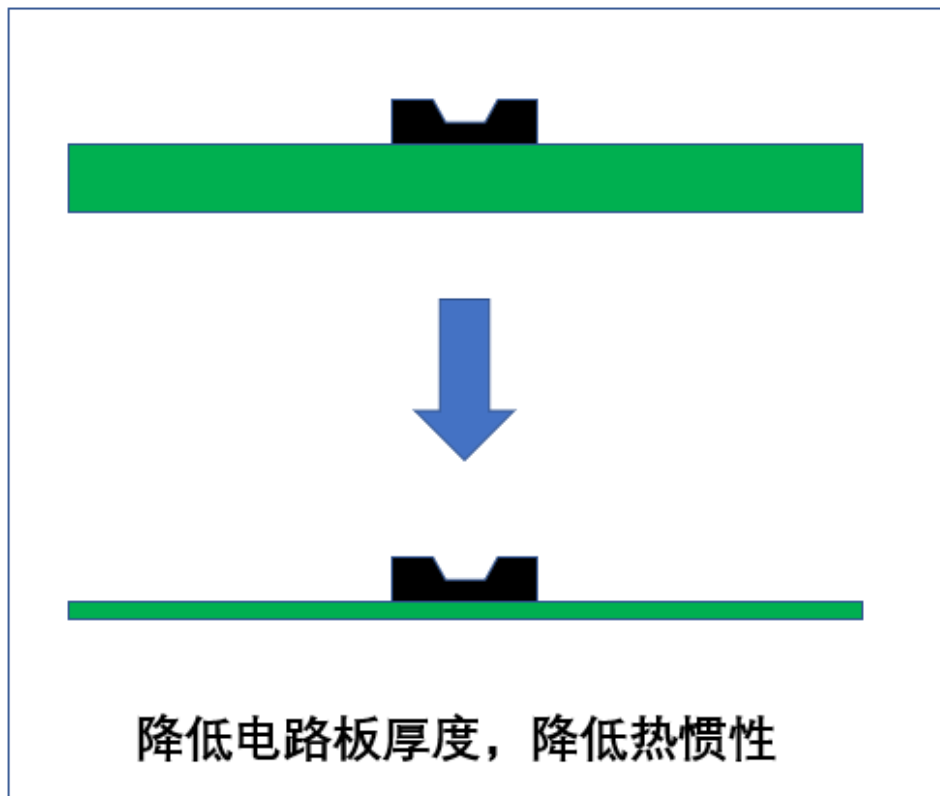


图5.5 降低电路板厚度

温湿度传感器NSHT30使用设计指南

2)采用独立的小板，通过插针连接到主板；

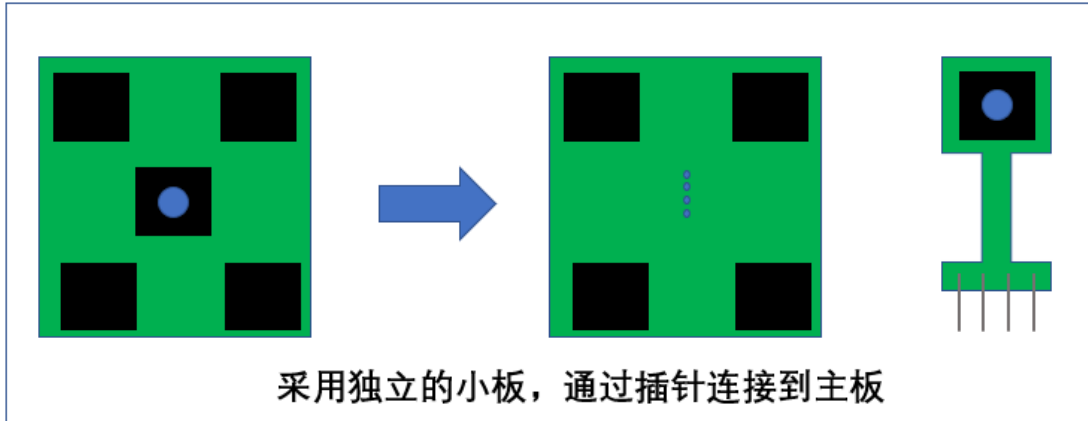


图5.6 采用独立小板

3)采用FPC软板，连接至主板；

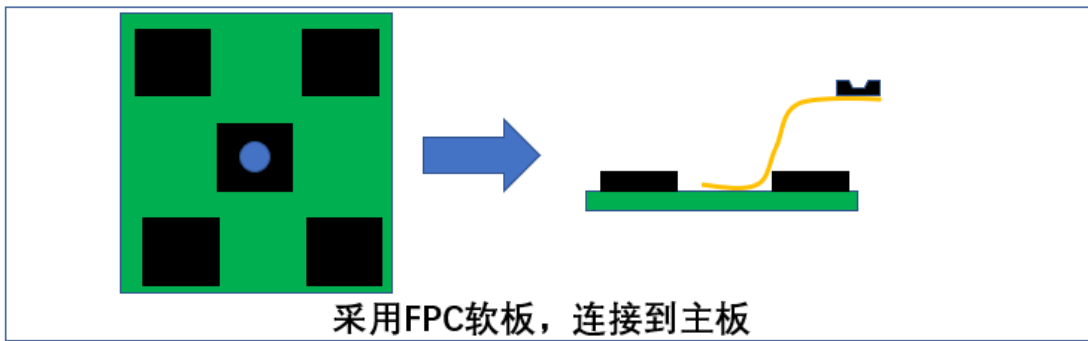


图5.7 采用FPC软板

温湿度传感器NSHT30使用设计指南

3.降低其它区域温度对传感器所在位置温度的影响；以降低温度影响为前提；几个基本方法如下：

- 1)装配上，湿度芯片与主板上热源间形成物理隔离，降低经空气热传导以及热辐射影响；
- 2)传感器尽量避免阳光直射，降低经热辐射路径传递的热量导致温度上升；
- 3)传感器尽量安装在独立在子板上；如必须安装在主板上，芯片位置与热源间尽量远离；主板上，在湿度芯片附近开孔；降低热源经过电路板上的热传导对芯片温度带来的影响；

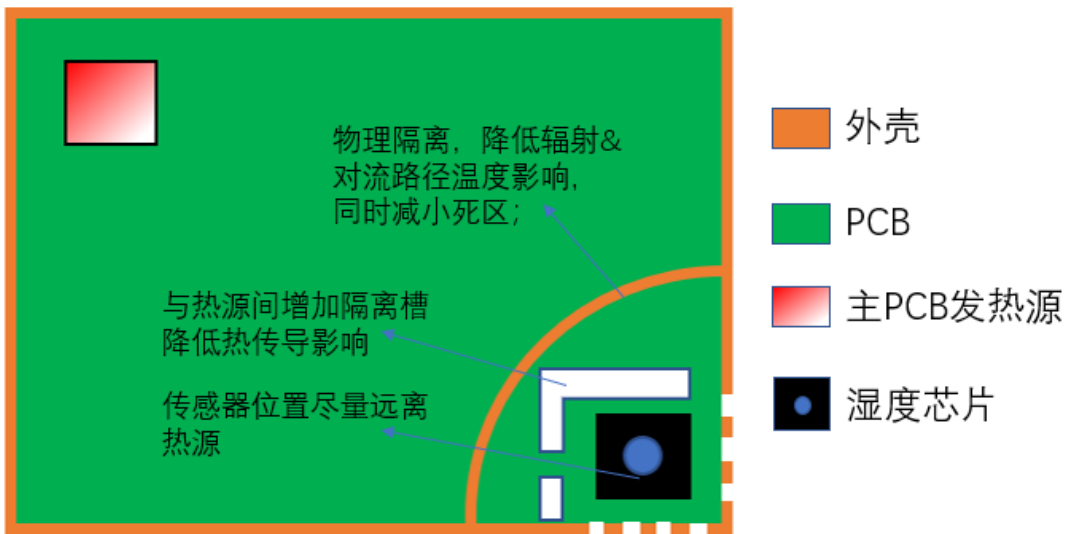


图5.8传感器区域热质量及受电路板热传导影响程度

温湿度传感器NSHT30使用设计指南

修订历史

版本	描述	作者	日期
1.0	创建	Juanjuan Shao	2021/08/23

销售联系方式: sales@novosns.com; 获取更多信息: www.novosns.com

重要声明

本文件中提供的信息不作为任何明示或暗示的担保或授权，包括但不限于对信息准确性、完整性，产品适销性、某特定用途方面的适用性或不侵犯任何第三方知识产权的陈述或保证。

客户应对其使用纳芯微的产品和应用自行负责，并确保应用的安全性。客户认可并同意：尽管任何应用的相关信息或支持仍可能由纳芯微提供，但将在产品及其产品应用中遵守纳芯微产品相关的所有法律、法规和相关要求。

本文件中提供的资源仅供经过技术培训的开发人员使用。纳芯微保留对所提供的产品和服务进行更正、修改、增强、改进或其他更改的权利。纳芯微仅授权客户将此资源用于开发所设计的整合了纳芯微产品的相关应用，不视为纳芯微以明示或暗示的方式授予任何知识产权许可。严禁为任何其他用途使用此资源，或对此资源进行未经授权的复制或展示。如因使用此资源而产生任何索赔、损害、成本、损失和债务等，纳芯微对此不承担任何责任。

有关应用、产品、技术的进一步信息，请与纳芯微电子联系（www.novosns.com）。

苏州纳芯微电子股份有限公司版权所有