

7 SMART Sensor Module 传感器模块

产品概述

7 SMART Sensor Module 传感器模块由电化学传感器和数据采集处理板组成。传感器为标准 7 系结构（盛密科技电化学传感器），数据采集处理板对传感器信号进行放大、采样、滤波，经 MCU 数据处理后实现气体的检测。通过配置不同的传感器，模块可对环境中存在的各类有毒有害及可燃性气体进行实时浓度检测，方便用户在不同场合下以简洁的方式快速组成系统，适用于室内、室外的空气质量检测，以及工业领域等的气体检测。

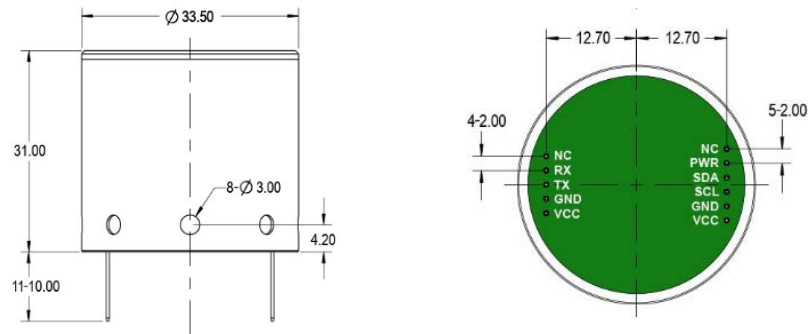
产品特点

- 统一的电气接口、机械尺寸和通讯协议；
- 可以根据需求选择不同的传感器，灵活应用于不同的场合；
- 模块内置温度、湿度传感器，以及对应的补偿算法，以减小环境温度湿度变化对测量结果的影响；
- 多种输出接口，包括 USART、I²C；
- 模块内置电源开关，可编程控制模块工作的启停；
- I²C 地址可编程，方便用户根据需求自定义。

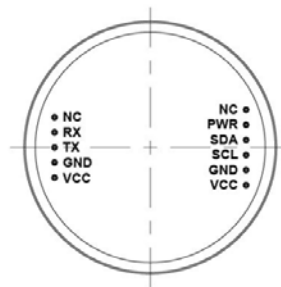
技术参数

传感器配置	7E4 电化学传感器	安装方式	11 脚插针
检测原理	电化学	工作温度	-20°C ~ +55°C
测量范围	由所选传感器而定，详见数据手册	工作湿度	(0%~90%)RH (无冷凝)
分辨率	由所选传感器而定，详见数据手册	工作压力	1 个标准大气压 ±10%
测量误差	由所选传感器而定，详见数据手册	外壳材质	铝合金
工作电压	(3.2~5.5) VDC	外形尺寸	Φ33.5x31mm
工作电流	≤ 5mA @5V	重量	30 克
信号输出方式	USART(3.3V TTL 电平) I ² C(3.3V TTL 电平)	工作寿命	传感器寿命见数据手册 数据板寿命 5 年 (无腐蚀环境下)

产品尺寸



引脚定义图



引脚定义：

1	NC	预留针脚（悬空）
2	RX	串口输入 RX
3	TX	串口输出 TX
4	GND	电源地
5	VCC	电源（3.2~5.5V）
6	NC	预留针脚（悬空）
7	PWR	模块电源使能（低电平关闭，高电平开启，内置上拉电阻）
8	SDA	I2C 信号 SDA（内置 10kΩ 上拉电阻）
9	SCL	I2C 信号 SCL（内置 10kΩ 上拉电阻）
10	GND	电源地
11	VCC	电源（3.2~5.5V）

注：两个 VCC 信号内部连通

USART 通信协议

1. 串行通信参数

起始位：1； 数据位：8； 停止位：1； 校验位：无； 波特率：115200bps；

2. 帧格式：(每一通信帧的格式如下)

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
H	ID	F	A	N	D	CRC16

H：数据头，1Byte 固定为 0x3A

ID：设备码，1Byte 默认为 0x10，可由用户自定义

F：功能码，1Byte，例如 (0x03)

A：起始地址，2Byte，例如 0x0001

N：数据长度，1Byte，以 2 个字节为单位，例如 (0x02: 4 字节)

D：数据，N*2Byte，高位在前，例如 (MSB LSB) 定义为有符号短整型 (signed short)

CRC16：数据校验，2Byte，使用 MODBUS_CRC16 校验算法 (算法详见附录 1)

3. 指令说明

3.1 读取传感器数据(单位为 ug/ m³)

上位机发送请求

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0000	0x01	0x0000	0x8352

模块接收正确数据应答

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0000	0x01	D	CRC16

D：接收到的数据，2Byte 高位在前

例：3A 10 03 00 00 01 00 6C 83 7F

传感器值 (ug/ m³): 00 6C 即 108ug/ m³

3.2 读取传感器数据(单位为 ppb)

上位机发送请求

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0001	0x01	0x0000	0x82AE

模块接收正确数据应答

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0001	0x01	D	CRC16

D : 接收到的数据, 2Byte 高位在前

例 : 3A 10 03 00 01 01 00 17 C2 A0

传感器值 (ppb): 00 17 即 23ppb

3.3 读取温度传感器数据(单位为°C)

上位机发送请求

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0002	0x01	0x0000	0x82EA

模块接收正确数据应答

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0002	0x01	D	CRC16

D : 接收到的数据, 2Byte 高位在前, 除以 100 后得到温度值, 例如 (D=0x0A3D =2621 得到温度为 26.21°C)

3.4 读取湿度传感器数据(单位为%RH)

上位机发送请求

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0003	0x01	0x0000	0x8316

模块接收正确数据应答

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0003	0x01	D	CRC16

D : 接收到的数据, 2Byte 高位在前, 除以 10000 后得到百分比湿度, 例如 (0x14 89 =5257 得到湿度为 52.57%)

3.5 读取多个参数 (地址 0000~0003)

上位机发送请求

首字节	设备码	功能码	起始地址	数据长度	数据	校验位
0x3A	0x10	0x03	0x0000	0x04	0x0000	0x9353

模块接收正确数据应答

首字节	设备码	功能码	起始地址	数据长度	数据	校验位

0x3A	0x10	0x03	0x0000	0x04	D	CRC16
------	------	------	--------	------	---	-------

D : 接收到的数据, 8Byte

例 : 3A 10 03 00 00 04 00 6C 00 5E 0A 19 14 CE C0 2F

其中 : 传感器值 (ug/m³) : 00 6C 传感器值 (ppb) : 005E 温度 : 0A 19 湿度 : 14 CE

3.6 校验错误应答

首字节	设备码	功能码	数据	校验位
0x3A	0x10	0x08	0x00	CRC16

例 : 3A1008000AF9

I²C 通信协议

1 . I²C 接口

参数	定义	状态	最小	最大	单位
f _{ck}	I2C 时钟频率	从模式		100	kHz
t _{su}	数据输入建立时间	从模式	6.5		ns
t _h	数据输入保持时间	从模式	15.5		ns

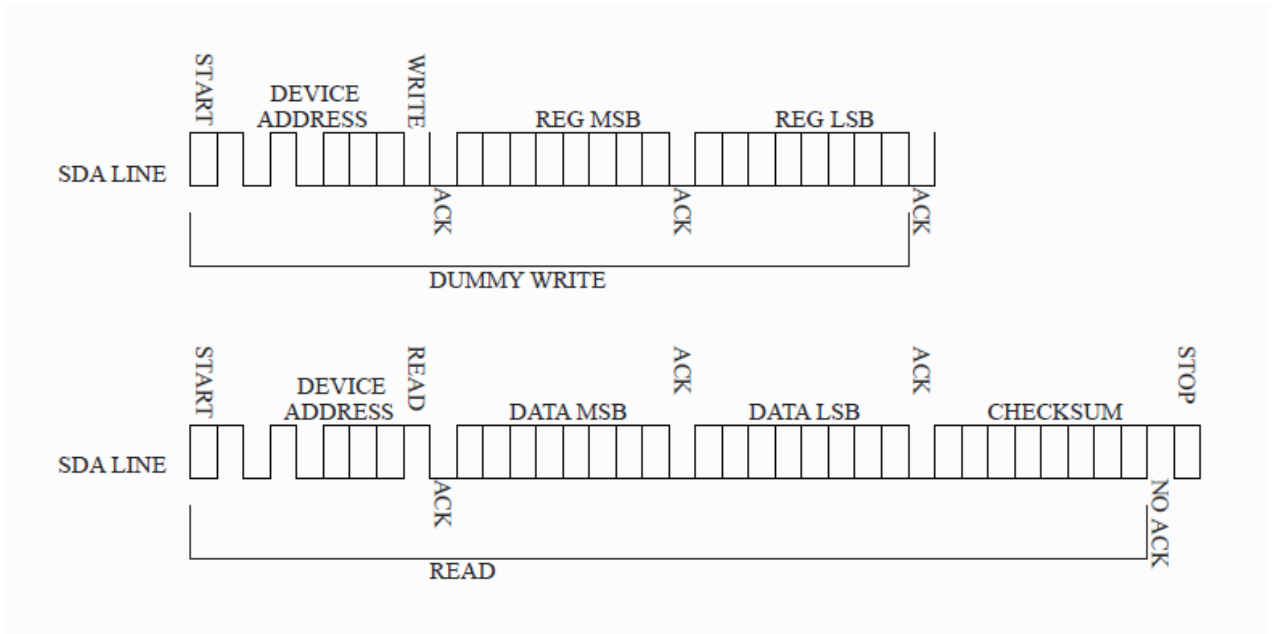
2 . 从设备地址

从设备地址可以通过软件工具自定义

默认设置如下 :

CO:	0	0	0	0	0	0	1	R/W
O3:	0	0	0	0	1	0	0	R/W
SO2:	0	0	0	0	1	0	1	R/W
NO2:	0	0	0	1	0	1	1	R/W

3. I²C 通信协议



CHECKSUM 为累加和取反校验，算法详见附录 2

4. 数据解析

REG 参数如下表，为模块中数据地址

REG	MSB	LSB
传感器值 (ug/m ³)	0x00	0x00
传感器值 (ppb)	0x00	0x01
温度	0x00	0x02
湿度	0x00	0x03

DATA 举例：

DATA	MSB	LSB	CHECKSUM	实际值	备注
传感器值 (ug/m ³)	0x00	0x05	0xFA	5ug/m ³	同 USART 数据转换方法
传感器值 (ppb)	0x00	0x05	0xFA	5ppb	
温度	0x0A	0x3D	0xB8	26.21°C	
湿度	0x14	0x89	0x62	52.57%	

注意事项

- 1) 本模块不具备电源反接保护及静电防护功能，用户在使用时请正确连接模块电源，并做好静电防护措施；
- 2) 请使用稳定的直流电源给模块供电，电源电压波动应小于1%；

附录1：MODBUS CRC16 算法

```
unsigned short modbus_CRC16(unsigned char *ptr, unsigned char len)
{
    unsigned short wrcrc=0xFFFF; //
    int i=0, j=0;
    for(i=0; i<len; i++)
    {
        wrcrc ^= *ptr++;
        for(j=0; j<8; j++)
        {
            if(wrcrc & 0X0001)
            {
                wrcrc = wrcrc >> 1 ^ 0XA001;
            }
            else
            {
                wrcrc >>= 1;
            }
        }
    }
    return wrcrc << 8 | wrcrc >> 8; //低位在前，高位在后
}
```

附录2：CHECKSUM 累加和校验

```
unsigned char CheckSum(unsigned char *buf, unsigned char len) //累加和校验值
{
    uint8_t i, ret = 0;
    for(i=0; i<len; i++)
    {
        ret += *(buf++);
    }
    ret = ~ret;
    return ret;
}
```