

• 产品描述

4SMART-C01传感器模块为四系列三电极电化学传感器搭建模块化检测平台，针对环境中存在的各类有毒有害气体进行实时浓度检测。模块以盛密科技电化学传感器为敏感元件(传感器具体技术参数详见相关数据手册)，通过信号采样/放大、滤波/转换完成气体浓度的检测。本模块内置温度补偿、线性校正等功能，具有良好的稳定性。本模块具有UART、DAC信号输出方式，可直接输出表征气体浓度数据的数字信号和模拟电压信号，操作简便，可以缩短用户后续开发周期。本模块具有零偏压断电自动短接PIN脚功能，对于零偏压传感器，上电数分钟后即可稳定使用；对于带偏压的传感器，建议用户上电24小时以上再进行使用。

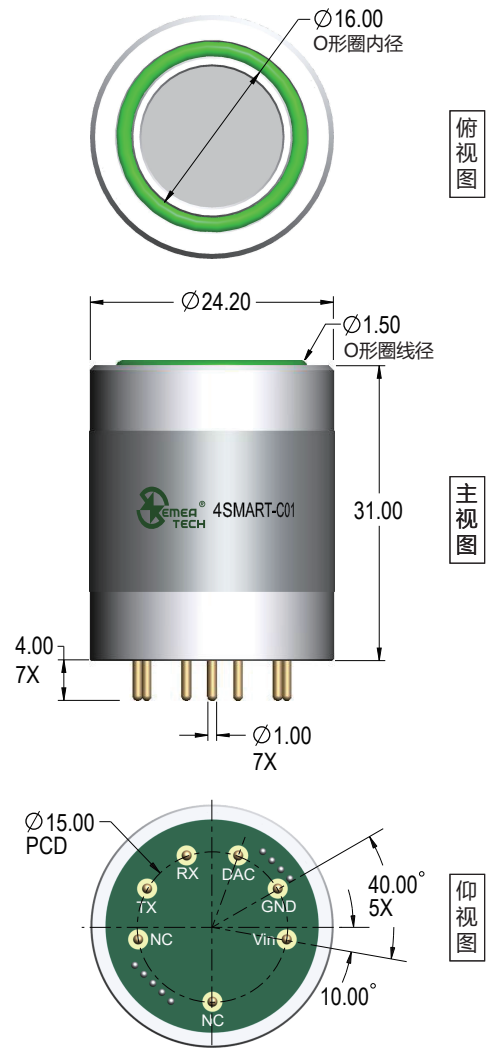
• 技术参数

产品型号:	4SMART-C01
检测气体:	有毒有害气体
检测原理:	电化学
量程:	详见数据手册
分辨率:	详见数据手册
测量误差:	< ±5%FS
工作电压:	(3.3 ~ 5.5) VDC
工作电流:	≤ 5 mA @ 3.3 VDC
输出方式:	UART (3.0V TTL电平) DAC(默认0.4 ~ 2.0VDC)
安装方式:	7脚插拔式
工作温度:	-20°C ~ 50°C
工作湿度:	0% ~ 90%RH (无冷凝)
工作压力:	1 ± 0.1 标准大气压
外壳材质:	铝合金
外形尺寸:	Φ 24.2 x 31.0 毫米
重量:	16 克

• 引脚定义

NC	悬空
VIN	电源输入正极
GND	电源输入负极
DAC	模拟电压输出
RX	串口接收
TX	串口发送
NC	悬空

• 产品尺寸



所有尺寸标注以毫米为单位
除非另有说明，所有公差±0.20毫米

• DAC(电压)信号输出

模块可通过“DAC”引脚输出表征气体浓度值的模拟电压信号。默认输出电压范围：0.4~2.0VDC。
(DAC信号需连接跟随器后，再输入后级信号采样系统)。

• 串行通信协议

波特率	9600 bps
数据位	8 位
停止位	1 位
校验位	无

本模块采用串口 (TXD/RXD) 进行数据传输，传输方式采用问答式。数据传输均为 16 进制 (HEX) 格式。

1. 终端读取模块信息命令

示例：AA 0F 01 C5 80 EE

- Byte1--AA: 命令起始符;
- Byte2--0F: 信息读取命令;
- Byte3--01: 模块地址 (默认为 0X01);
- Byte4--C5: CRC16 (Modbus) 校验高字节;
- Byte5--80: CRC16 (Modbus) 校验低字节;
- Byte6--EE: 命令结束符;

注：本命令中对 Byte2、Byte3 进行 CRC16 (Modbus) 校验

模块应答 (向终端发送信息数据)

示例：AA 0F 01 0F 00 14 00 05 00 02 00 01 02 C5 99 EE

- Byte1--AA: 命令起始符;
- Byte2--0F: 信息读取命令;
- Byte3--01: 模块地址;
- Byte4--0F: 传感器类型;
- Byte5/6--00/14: 模块测量范围 (16 进制);
- Byte7/8--00/05: 标定气体浓度 (16 进制);
- Byte9/10--00/02: 高报警点 (16 进制);
- Byte11/12--00/01: 低报警点 (16 进制);
- Byte13--02: 传感器读数单位 (%LEL: 0x00; %VOL: 0x01; PPM: 0x02; PPB: 0x03; 无: 0x04);
- Byte14--C5: CRC16 (Modbus) 校验高字节;
- Byte15--99: CRC16 (Modbus) 校验低字节;

Byte16--EE: 命令结束符;

注: 本命令中对 Byte2~Byte13 进行 CRC16 (Modbus) 校验

附表: 传感器类型代码

01 EX 02 CO 03 O2 04 H2 05 CH4 06 C3H8 07 CO2 08 O3 09 H2S 10 SO2 11 NH3 12 CL2
13 ETO 14 HCL 15 PH3 16 HBr 17 HCN 18 AsH3 19 HF 20 Br2 21 NO 22 NO2 23 NOX 24 CLO2
25 SiH4 26 无 27 无 28 无 29 无 30 无 31 THT 32 C2H2 33 C2H4 34 CH2O 35 无 36 无
37 C6H6 38 H2O2 39 C2H3CL 40 VOC 41 CH3SH 42 C4H8

例: AA 0F 01 0F 00 14 00 05 00 02 00 01 02 C5 99 EE (十六进制0F=十进制15, 即得到该传感器为PH3传感器)

2. 终端发送浓度数据读取命令

示例: AA 01 01 C1 E0 EE

Byte1--AA: 命令起始符;

Byte2--01: 数据读取命令;

Byte3--01: 模块地址;

Byte4--C1: CRC16 (Modbus) 校验高字节;

Byte5--E0: CRC16 (Modbus) 校验低字节;

Byte6--EE: 命令结束符;

注: 本命令中对 Byte2、Byte3 进行 CRC16 (Modbus) 校验

模块应答 (向终端发送浓度数据)

示例: AA 01 01 80 00 00 00 15 CA EE

Byte1--AA: 命令起始符;

Byte2--01: 数据读取命令;

Byte3--01: 模块地址;

Byte4--80: 数据符号位 (0x80: 负; 0x00: 正);

Byte5/6--00/00: 数据(ppm)整数部分 (0~65535);

Byte7--00: 数据(ppm)小数部分 (0.00~0.99);

Byte8--15: CRC16 (Modbus) 校验高字节;

Byte9--CA: CRC16 (Modbus) 校验低字节;

Byte10--EE: 命令结束符;

注: 本命令中对 Byte2~Byte7 进行 CRC16 (Modbus) 校验

3. 终端发送模块校零命令

示例: AA 02 01 C1 10 EE

Byte1--AA: 命令起始符;

Byte2--02: 校零命令;

Byte3--01: 模块地址;

Byte4--C1: CRC16 (Modbus) 校验高字节;

Byte5--10: CRC16 (Modbus) 校验低字节;

Byte6—EE: 命令结束符;

注: 1) 本命令中对 Byte2、Byte3 进行 CRC16 (Modbus) 校验;

2) 校零期间 LED 以 1 秒/次的频率闪烁, 持续时间 30 秒;

校零成功, 模块发送:

AA 02 01 10 D0 5C EE

校零失败, 模块发送:

AA 02 01 20 D0 48 EE

Byte1--AA: 命令起始符;

Byte2--02: 校零命令;

Byte3--01: 模块地址;

Byte4—10/20: 成功/失败标志;

Byte5—D0: CRC16 (Modbus) 校验高字节;

Byte6—5C/48: CRC16 (Modbus) 校验低字节;

Byte7—EE: 命令结束符;

注: 本命令中对 Byte2、Byte3、Byte4 进行 CRC16 (Modbus) 校验

4. 终端发送模块标定命令

示例: AA 03 01 C0 80 EE

Byte1--AA: 命令起始符;

Byte2--03: 标定命令;

Byte3--01: 模块地址;

Byte4--C0: CRC16 (Modbus) 校验高字节;

Byte5--80: CRC16 (Modbus) 校验低字节;

Byte6—EE: 命令结束符;

注: 1) 本命令中对 Byte2、Byte3 进行 CRC16 (Modbus) 校验

2) 标定期间 LED 以 1 秒/次的频率闪烁, 持续时间 120 秒;

标定成功, 模块发送:

AA 03 01 10 81 9C EE

标定失败, 模块发送:

AA 03 01 20 81 88 EE

Byte1--AA: 命令起始符;

Byte2--03: 标定命令;

Byte3--01: 模块地址;

Byte4—10/20: 成功/失败标志;

Byte5—81/81: CRC16 (Modbus) 校验高字节;

Byte6—9C/88: CRC16 (Modbus) 校验低字节;

Byte7—EE: 命令结束符;

注: 本命令中对 Byte2、Byte3、Byte4 进行 CRC16 (Modbus) 校验

5. 终端修改模块地址命令

示例: AA 04 02 C82 B1 EE

Byte1--AA: 命令起始符;

Byte2--04: 修改地址命令;

Byte3--02: 模块新地址;

Byte4--82: CRC16 (Modbus) 校验高字节;

Byte5--B1: CRC16 (Modbus) 校验低字节;

Byte6--EE: 命令结束符;

注: 本命令中对 Byte2、Byte3 进行 CRC16 (Modbus) 校验

地址修改成功, 模块发送:

示例: AA 04 02 10 30 AD EE

Byte1--AA: 命令起始符;

Byte2--04: 修改地址命令;

Byte3--02: 模块新地址;

Byte4--10: 地址修改成功标志;

Byte5--30: CRC16 (Modbus) 校验高字节;

Byte6--AD: CRC16 (Modbus) 校验低字节;

Byte7--EE: 命令结束符;

注: 本命令中对 Byte2~Byte4 进行 CRC16 (Modbus) 校验

6. 终端发送修改模块标气浓度命令

示例: AA 05 01 01 F4 51 3F EE

Byte1--AA: 命令起始符;

Byte2--05: 修改标准气体浓度命令;

Byte3--01: 模块地址;

Byte4/5--01/F4: 需要设置的标准气体浓度 (此处为 16 进制, 0x1F4)

Byte6--51: CRC16 (Modbus) 校验高字节;

Byte7—3F: CRC16 (Modbus) 校验低字节;

Byte8—EE: 命令结束符;

注: 本命令中对 Byte2~Byte5 进行 CRC16 (Modbus) 校验

修改成功, 模块发送:

AA 05 01 10 01 F4 E8 2E EE

修改失败, 模块发送:

AA 05 01 20 01 F4 E8 21 EE

Byte1--AA: 命令起始符;

Byte2--05: 修改标准气体浓度命令;

Byte3--01: 模块地址;

Byte4—10/20: 修改成功/失败标志;

Byte5/6—01/F4: 需要设置的标准气体浓度 (此处为 16 进制, 0x1F4)

Byte7—E8/E8: CRC16 (Modbus) 校验高字节;

Byte8—2E/21: CRC16 (Modbus) 校验低字节;

Byte9—EE: 命令结束符;

注: 本命令中对 Byte2~Byte6 进行 CRC16 (Modbus) 校验

• 注意事项

- 1) **未经国家法律、法规许可及认证, 不可将本模块用于具有防爆安全要求的场所及领域;**
- 2) 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;
- 3) 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 wcrc=0xFFFF; //
    int i=0, j=0;
    for (i=0; i<len; i++)
    {
        wcrc^=*ptr++;
        for (j=0; j<8; j++)
        {
            if (wcrc&0X0001)
            {
                wcrc=wcrc>>1^0XA001;
            }
            else
            {
                wcrc>>=1;
            }
        }
    }
    return wcrc<<8| wcrc>>8; //低位在前, 高位在后
}
```

CRC-16/MODBUS 算法:

在CRC计算时只用8个数据位, 起始位及停止位, 如有奇偶校验位也包括奇偶校验位, 都不参与CRC计算。

CRC计算方法是:

- 1、加载一值为0xFFFF的16位寄存器, 此寄存器为CRC寄存器。
- 2、把第一个8位二进制数据 (即通讯信息帧的第一个字节) 与16位的CRC寄存器的相异或, 异或的结果仍存放于该CRC寄存器中。
- 3、把CRC寄存器的内容右移一位, 用0填补最高位, 并检测移出位是0还是1。
- 4、如果移出位为零, 则重复第三步 (再次右移一位); 如果移出位为1, CRC寄存器与0XA001进行异或。
- 5、重复步骤3和4, 直到右移8次, 这样整个8位数据全部进行了处理。
- 6、重复步骤2和5, 进行通讯信息帧下一个字节的处理。
- 7、将该通讯信息帧所有字节按上述步骤计算完成后, 得到的16位CRC寄存器的高、低字节进行交换。
- 8、最后得到的CRC寄存器内容即为: CRC校验码。
- 9、举例: 一条命令 05 01 01 F4 通过上述程序, 得到wcrc返回值为51 3F, 作为校验码, 这样我们就得到了通气标定命令: AA 05 01 01 F4 51 3F EE。