

• 产品描述

盛密科技4LEL-SMART可燃气体传感器模块利用催化燃烧工作原理，对环境中的可燃性气体浓度进行检测。模块以盛密科技4LEL-3.0V催化燃烧传感器为敏感元件，具有良好的稳定性。模块与接收终端采用四线制连接方式，将表征气体浓度数值的数字信号通过UART总线输出，操作简便，利于用户进行二次开发。传感器具体技术参数可详见数据手册。

• 技术参数

产品型号:	4LEL-SMART
检测气体:	天然气、液化气等可燃性气体
检测原理:	催化燃烧
量程:	0 ~ 100% LEL
分辨率:	1% LEL
测量误差:	< ±5%FS
工作电压:	(3.3 ~ 3.6) VDC
工作电流:	≤ 110 mA @ 3.3 VDC
输出方式:	UART (3.3V TTL电平)
工作温度:	0°C ~ 40°C
工作湿度:	≤ 98%RH (25°C)
工作压力:	86 ~ 116 KPa
存储温度:	-20°C ~ 40°C
外形尺寸:	Φ 24.2 x 26.8 毫米
重量:	25 克

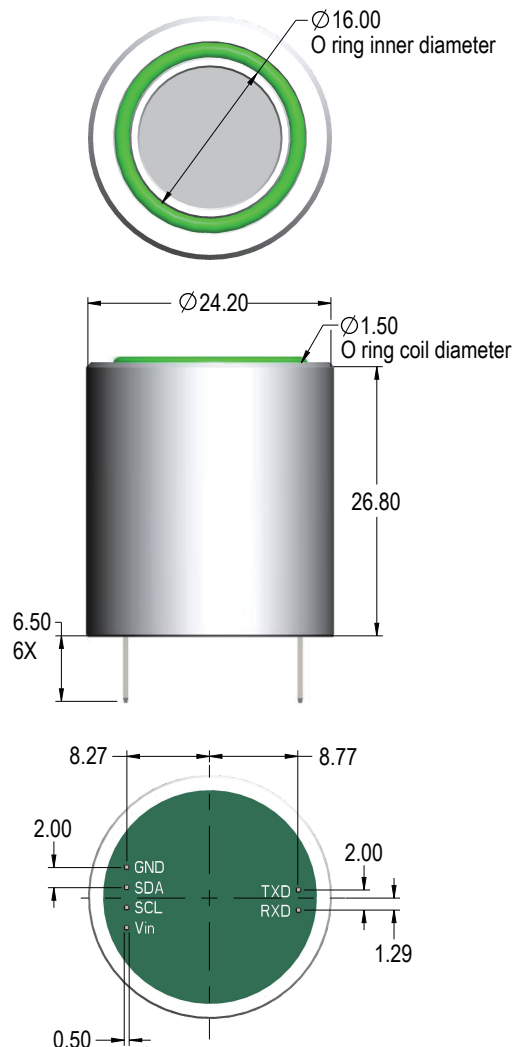
• 引脚定义

Vin	电源输入正极
GND	电源输入负极
TXD	串口发送
RXD	串口接收

• 通讯设置

波特率:	9600 bps
数据位:	8 位
停止位:	1 位
校验位:	无

• 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

• 通讯命令

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

命令起始符	信息读取命令	模块地址 (默认为0x01)	CRC16(Modbus) 校验高字节	CRC16(Modbus) 校验低字节	命令结束符
AA	0F	01	C5	80	EE

例: AA 0F 01 C5 80 EE

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)

命令起始符	信息读取命令	模块地址	传感器类型 (LEL)	模块测量范围 (16进制)	模块测量范围 (16进制)
AA	0F	01	05	00	64
标定气体浓度 (16进制)	标定气体浓度 (16进制)	高报警点 (16进制)	高报警点 (16进制)	低报警点 (16进制)	低报警点 (16进制)
00	19	00	19	00	0F
读数单位	CRC16(Modbus) 校验高字节	CRC16(Modbus) 校验低字节	命令结束符		
00	63	84	EE		

Byte13 - 00: 传感器读数单位 (%LEL: 0x00; %VOL: 0x01; PPM: 0x02; PPB: 0x03; 无: 0x04)

例: AA 0F 01 05 00 64 00 19 00 19 00 0F 00 63 84 EE

注: 本命令中对Byte2~Byte13进行CRC16(Modbus)校验

2. 终端发送浓度数据读取命令

命令起始符	数据读取命令	模块地址	CRC16(Modbus) 校验高字节	CRC16(Modbus) 校验低字节	命令结束符
AA	01	01	C1	E0	EE

例: AA 01 01 C1 E0 EE

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)

命令起始符	数据读取命令	模块地址	数据符号位	传感器读数(LEL) 整数部分	传感器读数(LEL) 整数部分
AA	01	01	80	00	00
传感器读数 小数部分	CRC16(Modbus) 校验高字节	CRC16(Modbus) 校验低字节	命令结束符		
00	15	CA	EE		

Byte4 - 80: 数据符号位 (0x80: 负; 0x00: 正);

Byte5/6 - 00/00: 传感器读数(LEL)整数部分 (0 ~ 65535);

Byte7 - 00: 传感器读数小数部分 (0.00 ~ 0.99);

例: **AA 01 01 80 00 00 00 15 CA EE**

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

命令起始符	校零命令	模块地址	CRC16(Modbus) 校验高字节	CRC16(Modbus) 校验低字节	命令结束符
AA	02	01	C1	10	EE

例: **AA 02 01 C1 10 EE**

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

命令起始符	校零命令	模块地址	成功/失败标志	CRC16(Modbus) 校验高字节	CRC16(Modbus) 校验低字节	命令结束符
AA	02	01	10/20	D0	5C/48	EE

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

命令起始符	标定命令	模块地址	CRC16 (Modbus) 校验高字节	CRC16 (Modbus) 校验低字节	命令结束符
AA	03	01	C0	80	EE

例: **AA 03 01 C0 80 EE**

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

命令起始符	标定命令	模块地址	成功/失败标志	CRC16(Modbus) 校验高字节	CRC16(Modbus) 校验低字节	命令结束符
AA	03	01	10/20	81/81	9C/88	EE

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

命令起始符	修改地址命令	模块新地址	CRC16 (Modbus) 校验高字节	CRC16 (Modbus) 校验低字节	命令结束符
AA	04	02	82	B1	EE

例：**AA 04 02 82 B1 EE**

注：本命令中对Byte2、Byte3进行CRC16(Modbus)校验

地址修改成功，模块发送：

AA 04 02 10 30 AD EE

命令起始符	修改地址命令	模块新地址	地址修改 成功标志	CRC16(Modbus) 校验高字节	CRC16(Modbus) 校验低字节	命令结束符
AA	04	02	10	30	AD	EE

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

命令起始符	修改标准气体 浓度命令	模块地址	需要设置的 标准气体浓度	需要设置的 标准气体浓度	CRC16(Modbus) 校验高字节
AA	05	01	00/32	00/32	D0
CRC16(Modbus) 校验低字节	命令结束符				
FD	EE				

Byte4/5 - 00/32: 需要设置的标准气体浓度 (此处为16进制, 0x032)

例: **AA 05 01 00 32 D0 FD EE**

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 00 32 69 EC EE

修改失败, 模块发送:

AA 05 01 20 00 32 69 E3 EE

命令起始符	修改标准气体浓度命令	模块地址	修改成功/失败标志	需要设置的标准气体浓度	需要设置的标准气体浓度
AA	05	01	10/20	00/32	00/32
CRC16(Modbus)校验高字节	CRC16(Modbus)校验低字节	命令结束符			
69/69	EC/E3	EE			

Byte5/6 - 00/32: 需要设置的标准气体浓度 (此处为16进制, 0x032)

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