



产品规格书

产品名称:	二氧化碳传感器
产品型号:	MQC02-1
版本:	V1.0
制定人:	贺根文
审核人:	袁超





修改记录表:

版本	更改内容	更改人	更改日期
V1.0	新建	贺根文	2020-4-24



一、产品描述

MQCO2-1是一款基于非分光带红外吸收原理的二氧化碳传感器，能够实时检测空气中的二氧化碳浓度，具有自校准、体积小、性能优越，一致性好，无氧气依赖性等特点。本产品内置温度补偿，同时具有串口输出、PWM输出两种通讯方式，方便使用。



本产品广泛应用于空气质量检测仪、空气净化设备、新风系统、空调控制、智能家居、物联网环境信息采集、农业生产、冷链运输等相关领域。

二、技术指标

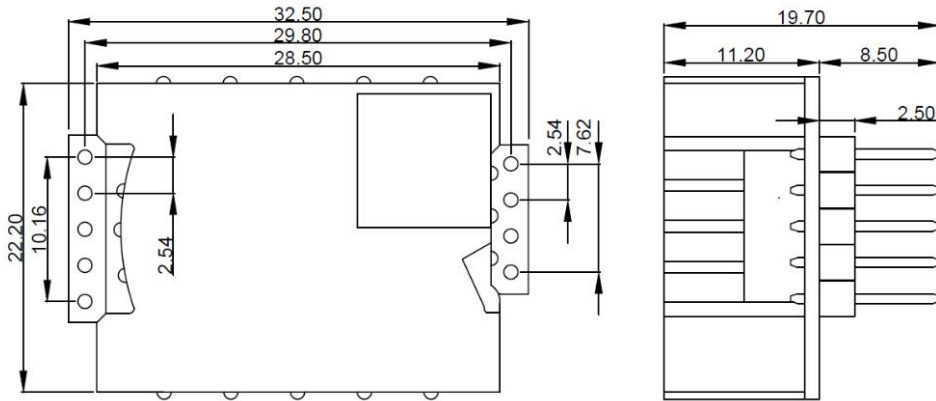
量程范围	400~5000ppm
精度	±(50ppm+5%*读数)
通讯接口	URAT_TTL(3.3V) PWM(3.3V)
分辨率	1ppm
响应时间	小于20s
数据更新时间	4s
预热时间	小于25s(可操作) 小于2min(90%精度) 小于10min(最大精度)
供电电压	DC(5.0±0.5)V
工作电流	平均小于70mA, 峰值小于150mA
工作条件	-10℃~+50℃, 0~95%RH, 无凝结
存储条件	-30~+70℃, 0~95%RH, 无凝结
产品寿命	大于5年
产品尺寸	32.5mm×22.2mm×19.7mm(L*W*H)



针脚间距

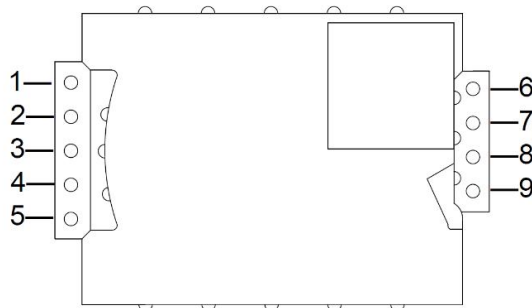
2.54mm

三、外形及结构尺寸



32.5mm×22.2mm×19.7mm(L*W*H)，公差±0.3mm

四、接口定义



序号	名称	描述
1	NC	预留
2	RX	串口端（电路板串口接收端）
3	TX	串口端（电路板串口发送端）
4	NC	预留
5	NC	预留
6	+5V	电源输入端（+5V 端）
7	GND	电源输入端（接地端）
8	NC	预留
9	PWM	脉宽调制



五、串口通信协议

1. 串口配置：9600 波特率，8 数据位，1 停止位，无奇偶校验；

2. 接收主机命令方式：

1) 接收主机命令：FE 04 00 03 00 01 D5 C5；

响应回复数据：FE 04 02 01 E5 6D 3F；其中FE 04 02代表响应数据帧头，01 E5，代表二氧化碳浓度485ppm，6D 3F为CRC校验数据；

2) 接收主机命令：64 69 03 5E 4E；

响应回复数据：64 69 03 01 0A 02 00 00 00 00 00 9B F0；其中64 69 03 01代表响应数据帧头，0A 02代表二氧化碳浓度，02为高位，0A为低位，即20A，表示522ppm，9B F0为CRC校验数据；

3) 接收主机命令：11 01 01 ED

响应回复数据：16 05 01 02 06 00 00 DC；其中FE 16 05 01代表响应数据帧头，02 06代表二氧化碳浓度518ppm；

3. CRC计算

```
uint16_t CO2ModbusComm::modbus_calcuCRC(uint8_t *dataarray, uint16_t datalen)
{
    uint8_t uchCRCHi = 0xFF ; /* CRC 的高字节初始化*/
    uint8_t uchCRCLo = 0xFF ; /* CRC 的低字节初始化*/
    uint16_t uIndex ; /* CRC 查询表索引*/
    uint16_t crc;
    const uint8_t auchCRCHI[] = {
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,
        0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
        0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,
        0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
        0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,
        0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
```



```
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,  
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,  
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,  
0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,  
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,  
0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,  
0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,  
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,  
0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,  
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,  
0x40  
};
```

```
const uint8_t auchCRCLo[] = {  
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4,  
0x04, 0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,  
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,  
0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,  
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,  
0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,  
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,  
0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,  
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,  
0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,  
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,  
0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,  
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,  
0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,  
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,  
0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,  
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,  
0x40  
};
```

```
while (datalen--)* 完成整个报文缓冲区*/  
{  
    uIndex = uchCRCLo ^ *dataArray++;/* 计算CRC */  
    uchCRCLo = uchCRCHI ^ auchCRCHI[uIndex];
```



```

uchCRCHi = auchCRCLo[uIndex];
}

crc = (uint16_t)uchCRCHi *256;
crc += (uint16_t)uchCRCLo;
return crc;

```

六、PWM计算方式

周期	1004ms ± 5%
中部周期	1000ms ± 5%
通过 PWM获得当前CO ₂ 浓度值的计算公式： $C_{ppm} = 5000 \times (T_H - 2ms) / (T_H + T_L - 4ms)$	
<p>C_{ppm} 为通过计算得到的 CO₂浓度值单位为 ppm T_H 为一个输出周期中输出为高电平的时间 T_L 为一个输出周期中输出为低电平的时间</p>	



七、注意事项

- 本产品采用 NDIR 技术，灯泡点亮的时间段内需要的电流比较大，建议电源电流大于 300mA。
- 从结露状态进入非结露状态，2 小时内系统可以恢复到正常工作状态。
- 请尽量避免震动，如果经过震动、跌落、焊接时间长可能会使传感器的测量准确性变差。如果发生严重的震动或跌落情况后，需要验证产品的准确度。
- 不要在粉尘密度大的环境长期使用。
- 避免在含有 HF, H₂S, SO₂, HCL, NOX, NH₃, PH₃, CL₂, F₂, O₃, H₂O₂ 等酸性、碱性、强氧化性气体环境中长期使用。
- 避免对外壳施加任何方向的外力，会影响传感器性能，并可能造成损坏
- 请勿直接触摸，确保操作人员佩戴静电释放手环。
- 本产品外壳没有接地，对 ESD 防护要求比较高的应用，请联系厂家技术支持。