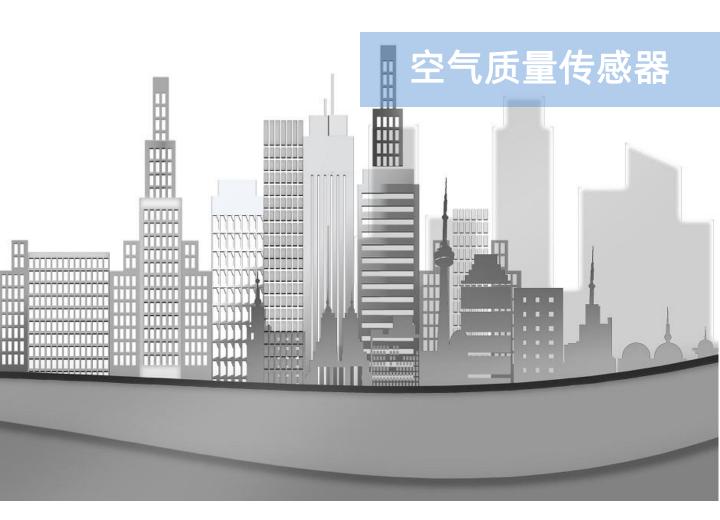


# **Product Catalog**



# 上海迅饶自动化科技有限公司

地址:上海市闵行区七宝镇中春路8633弄

(万科虹桥云)25幢603室

电话: 021-58776098

Email: sales@opcmaster.com

网址1: www.bacnetchina.com

网址2: www.opcmaster.com







如需了解更多迅饶产品,欢迎关注公众号: BACnet



# 空气质量传感器产品目录



# 空气质量传感器

	技术指标					
工作温度范围	0°C ~ +50°C					
工作湿度范围	0%RH~95%RH(无凝露)					
储运温度范围	-30°C ~ +50°C					
电源输入	DC 12V~24V(串口为5V DC)					
平均工作电流	≤80mA					
信号输出	Modbus-RTU,起始位1位,数据位8位,停止位无, 无校验/偶检验,波特率9600bps(默认)					
预热时间	2分钟(甲醛和TVOC需要预热稳定,其它参数上电即显示)					
物理接口	PH 2.0 端子座					
安装方式	吸顶安装					
产品重量	200g					
产品尺寸	100mm×100mm×38.5mm (L×W×H)					

检测项目	检测范围	检测分辨率	检测精度
CO <sub>2</sub>	400ppm ~ 5000ppm 1ppm		±(50ppm+读数的5%)
甲醛	0ug/m³ ~ 1000ug/m³	1ug/m³	±25%FS
TVOC	0ug/m³ ~ 2000ug/m³	1ug/m³	±25%FS
PM2.5	0ug/m³ ~ 1000ug/m³	1ug/m³	±10%FS
PM10	0ug/m³ ~ 1000ug/m³	1ug/m³	±10%FS
温度	-20°C ~ +60°C	0.1℃	±1.0℃
湿度	5%RH ~ 95%RH	0.1%RH	±5%RH



#### 一、功能描述

IAQ-1007-S 是一款高性价比的数字输出传感器模块,采用UART和RS485输出模式,集 $CO_2$ 、甲醛、TVOC、激光粉尘PM2.5/PM10颗粒物、温度、湿度传感器于一体。可对所处环境进行实时全面检测,具有良好的稳定性,非常方便客户使用。

# 二、应用场景

IAQ-1007-S空气质量传感器主要应用在智能楼宇、商业办公楼、智能家居、酒店、医院、学校等大型建筑物内部,它可以实时检测 CO₂、甲醛、TVOC、PM2.5、 PM10、温度、湿度等空气指标数据,以满足人们对环境健康的需求。

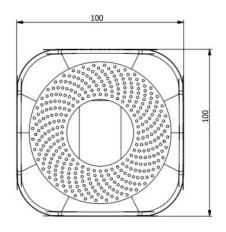


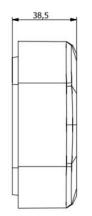
#### 三、主要特点

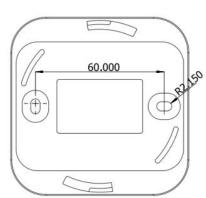
- ●可同时测量 CO<sub>2</sub>、甲醛、TVOC、PM2.5、 PM10、温度、湿度七组数据
- ●灵敏度高,数据稳定
- ●UART、RS485信号输出,通讯稳定
- ●温度分辨率0.1℃,湿度分辨率0.1%RH
- ●每隔1秒自动通过UART信号输出七组传感器监测数据



# 四、产品尺寸图



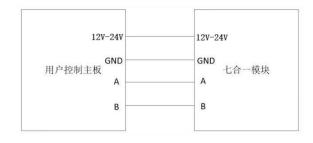




# 五、接线说明

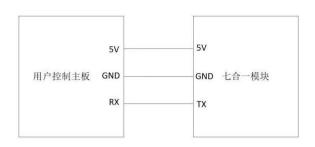
## 1. RS485

序号	名称	定义
1	В	RS485B
2	Α	RS485A
3	GND	电源地
4	VDD	接电源 12~24V DC



#### 2. UART

序号	名称	定义
1	TX	UART数 据输出 脚
2	RX	悬空
3	GND	电源地
4	5V	接电源 5V DC





#### 六、注意事项

- 1.初次上电使用需预热2分钟以上;
- 2.切勿拆解传感器,以防出现不可逆破坏;
- 3.请勿将传感器安装在强对流气体环境下使用;
- 4.将传感器安装在需要检测的位置,应远离热源;
- 5.请勿将传感器长时间置于高浓度有机气体及腐蚀性气体中;
- 6.避免将传感器放在盐雾、酸性类气体(二氧化硫、盐酸等)环境中;
- 7.避免地面沙尘, 飘絮物等大尘埃颗粒甚至絮状物污染导致传感器内的风扇缠绕阻转, 建议使用设备采取适当的预过滤处理;
- 8.本传感器适用于普通室内环境,使用时需满足以下条件:无导电尘埃、无腐蚀金属及破坏绝缘的 气体存在,避免直接光照和雨淋;
- 9.传感器在不同情况下长时间储存,传感器会发生可逆性漂移,使用前需要对传感器进行预热。 预热时间建议储存时间三个月以下时,预热12h以上;

储存时间三个月至六个月时, 预热24h以上;

储存时间六个月以上时, 预热48h以上;



附: UART通信协议

字节	名称	说明		
B0	帧头1			
B1	帧头2	固定值 92h		
B2	数据	CO₂高字节		
В3	数据	CO <sub>2</sub> 低字节		
B4	数据	甲醛高字节		
B5	数据	甲醛低字节		
В6	数据	TVOC 高字节		
В7	数据	TVOC 低字节		
B8	数据	PM2.5 高字节		
B9	数据	PM2.5 低字节		
B10	数据	PM10 高字节		
B11	数据	PM10 低字节		
B12	数据	温度整数部分		
B13	数据	温度小数部分		
B14	数据	湿度整数部分		
B15	数据     湿度小数部分			
B16	校验和	湿度校验和		

说明:校验和B16等于: B0+B1+B2+B3+······+B15的值,取低8位。 当温度的数据 B12的bit7=1 时,代表是负温度,B12 的bit7=0 时,是正温度。 例如,当 B12=9Bh 时,此时 bit7=1,表示是负温度,此时实际温度就是-27℃; 如果 B12=1Bh,此时 bit7=0,代表是正温度,此时实际温度就是 27℃。



#### 七、IAQ-1007-S 通讯协议

#### 1、概述

通讯协议详细地描述了环境空气质量传感器的输入和输出命令、信息和数据,以便第三方使用和开发。

#### 物理接口:

连接上位机的主通信口,采用标准串行RS-485通讯口。

信息传输方式为异步方式,起始位1位,数据位8位,停止位1位,无校验。

数据传输缺省速率为9600b/s。

#### 2、通讯协议详述

- 2.1 通讯方式:
- (1) 所有回路通信应遵照主/从方式。在这种方式下,信息和数据在单个主站和从站(监控设备)之间传递;
- (2) 支持广播模式。
- (3) 无论如何都不能从一个从站开始通信。
- (4) 若主站或任何从站接收到含有未知命令的包裹,则该包裹将被忽略,且接收站不予响应。
- 2.2 返回数据帧结构描述

每个数据帧组成如下: (RTU模式)

地址

功能代码

数据数量

数据1

. . .

数据n

CRC 16位校验

#### 3、传输格式

#### 3.1 主机发送读地址命令

地址	功能码	寄存器地址 高位	寄存器地址 低位	数据个数 高位	数据个数 低位	CRC 16位校验
00	03	01	00	00	01	xxxx低位在前

#### 从机返回地址:

地址	功能码	字节长度	地址高位	地址低位	CRC 16位校验
00	03	02	00	xx	xxxx低位在前

#### 示例:

主机发送: 00 03 01 00 00 01 84 27

从机响应: 00 03 02 00 address CRCL CRCH; 若address为2, 则读取的传感器地址为 2



# 七、IAQ-1007-S 通讯协议

# 3.2 主机发送读数据命令

地址	功能码	数据起始地址 高位	数据起始地 址低位	数据个数 高位	数据个数 低位	CRC 16位校验
00	03	00	00	00	07	低位在前

(**目前仅支持读取所有数据,从地址0002开始,读取7个数据值**),只有7个传感器数据,对应7个传感器 地址,**数据起始地址高位永远为0,返回数据个数高位永远为0**。

内部报文信息:

起始地址	数据个数	说明
只能是如下		
0x0002	1	CO₂浓度
0x0003	1	甲醛浓度
0x0004	1	TVOC 浓度
0x0005	1	PM2.5浓度
0x0006	1	PM10浓度
0x0007	1	温度值
0x0008	1	湿度值

起始地址只能用以上列表中的地址,否则不回应.数据个数如果超出,也不回应,比如,起始地址0X0007,可以返回的数据个数只可能是1或2,如果要回3个数据则不回应.同理,地址0X0008,返回数据个数只能是1.返回数据个数0,也不回应。

从机传感器返回值:

从机地址	功能码	数据个数	数据N	CRC
XX	03	XX	XX	xxxx

#### 字节长度仅指数据长度

N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13
CO <sub>2</sub>	CO <sub>2</sub>	甲醛	甲醛	TVOC	TVOC	PM2.5	PM2.5	PM10	PM10	温度	温度	湿度	湿度
高位	低位	高位	低位	高位	低位	高位	低位	高位	低位	高位	低位	高位	低位

示例: 主机发送 01 03 00 02 00 07 CRCL CRCH

响应: 01 03 0E 02 26 00 26 00 34 00 0E 00 11 01 1C 02 8A CRCL CRCH



# 七、IAQ-1007-S 通讯协议

注: 温湿度返回计算

返回温度16进制数据: 0x00FF,对应十进制255,表示温度为25.5℃ 返回湿度16进制数据: 0x0311,对应十进制785,表示湿度为78.5%RH

返回温度16进制数据: 0x8064,最高位为1表示负数,对应十进制-100,表示温度为-10.0℃

#### 3.3 设置地址命令

此命令可以设置从机地址,地址值1到255,此后,从机地址就是主机设置的地址。主机发送指令:

字节	名称	说 明
В0	广播地址	00h
B1	功能码	06h
B2	寄存器地址高	01h
В3	寄存器地址低	00h
B4	从机地址高	00h
B5	从机地址低	从机地址(1~255)
В6	CRC16	低
В7	CRC16	高

#### 从机应答:

字节	名称	说 明
В0	广播地址	00h
B1	功能码	06h
B2	字节数	02h
В3	从机地址高	00h
B4	从机地址低	从机地址(1~255)
B5	CRC16	低
В6	CRC16	高

示例: 00 06 00 00 address CRCL CRCH 从机应答: 00 06 02 00 address CRCL CRCH

若address==2,则设置的从机地址为2.

主机得到从机回应后,认为设置地址成功,以后首字节按照address设置的地址,发送读数据命令。



### 4、主机数据采样频率

读取温湿度传感器数据时,上位机读取数据每次间隔时间不小于500ms,推荐值1s。

# 5、CRC校验参考

```
#include "XXX.h"
                    //单片机头文件
unsigned int GetCRC16(unsigned char *ptr, unsigned char len)
{
  unsigned int index;
  unsigned char crch = 0xFF; //高CRC字节
  unsigned char crcl = 0xFF; //低CRC字节
  unsigned char code TabH[] = { //CRC高位字节值表
    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
      };
```



```
unsigned char code TabL[] = { //CRC低位字节值表
    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 (len--) //计算指定长度的CRC
    index = crch ^ *ptr++;
    crch = crcl ^ TabH[ index];
    crcl = TabL[ index];
  }
  return ((crch<<8) | crcl);
}
```