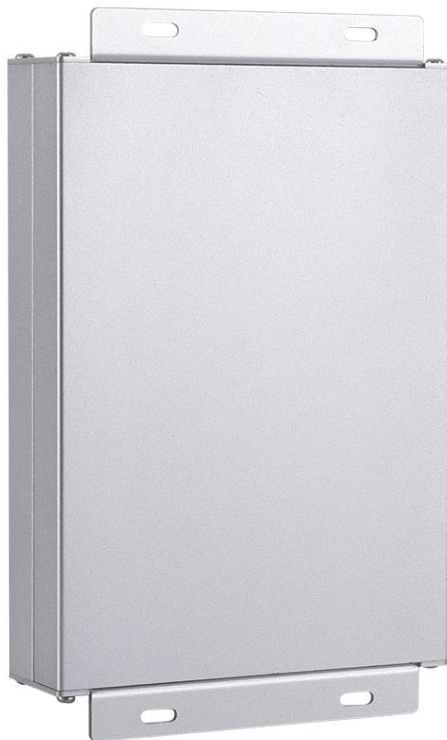




# 产品规格书

产品名称:	环境空气质量检测仪
产品型号:	MAQ01020406070914-S-P4
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

环境空气质量检测模组 MAQ01020406070914-S-P4 是一款数字信号输出的多参数传感器产品, 集温度、湿度、CO<sub>2</sub>、PM2.5、PM10、TVOC、甲醛检测于一体, 采用高性能的传感器及控制芯片, 具备精度高、稳定性好等特点。设备采用宽电压 DC12 ~ 24V 供电, RS485 信号输出, 标准 Modbus-RTU 通信协议, 通讯地址可设置, 波特率可更改。广泛应用于机车空调、楼宇暖通、建筑节能、智能家居、学校、医院、机场车站等场所。

## 二、功能特点

- 1、采用隔离的 RS485 电路, 通讯稳定。
- 2、采用铝合金壳体, 适用多种安装方式。
- 3、可同时测量温度、湿度、CO<sub>2</sub>、PM2.5、PM10、TVOC、甲醛多种参数, 且能扩展更多的空气质量参数。

## 三、技术指标

工作温度范围	0 ~ 50°C
工作湿度范围	0 ~ 95%RH (非结露)
储存温度范围	-40 ~ 50°C
电源输入	DC 12~24V
最大工作电流	≤100mA (DC24V)
信号输出	Modbus-RTU, 起始位 1 位, 数据位 8 位, 停止位 1 位, 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)
安装方式	吸顶安装、壁挂安装
产品重量	465 克



产品尺寸	188.0mm×114.5mm×33.0mm
------	------------------------

#### 四、检测参数

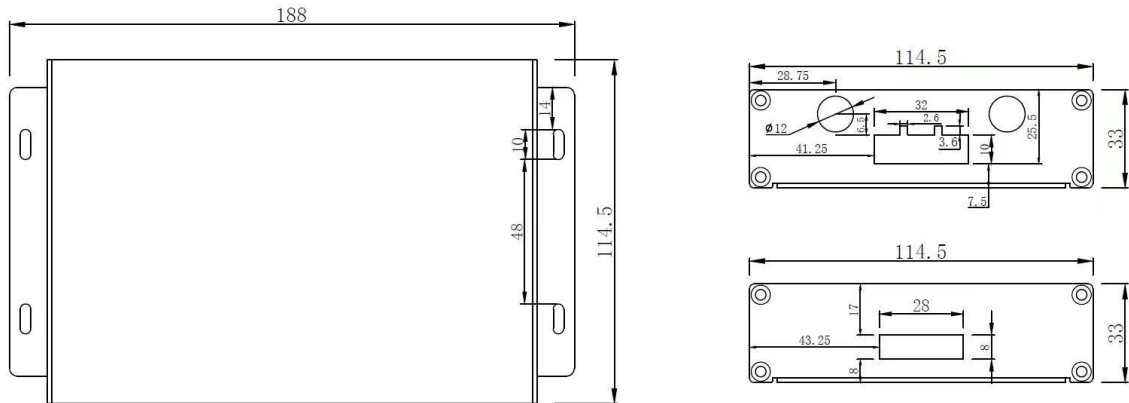
检测项目	检测范围	检测精度	检测原理
温度	-40~125°C	±0.5°C (@25°C) , ±1°C (全量程)	MEMS 器件
湿度	0~ 100%RH	±3%RH(@25°C, 30%~80%RH; 其它±5%RH)	MEMS 器件
CO <sub>2</sub>	400~5000ppm	± (75ppm + 5% of reading )	NDIR 非分散性红外 传感器
PM2.5	0~1000µg/m <sup>3</sup>	±10%(@100~500µg/m <sup>3</sup> ); ±10µg/m <sup>3</sup> (@ 0~100µg/m <sup>3</sup> )	激光散射
PM10	0~1000µg/m <sup>3</sup>	±10%(@100~500µg/m <sup>3</sup> ); ±10µg/m <sup>3</sup> (@ 0~100µg/m <sup>3</sup> )	激光散射
TVOC	0~2000ppb	±30%FS@25°C	半导体气体传感器
甲醛	0~1000ppb	±15%FS@25°C	燃料电池传感器

备注:

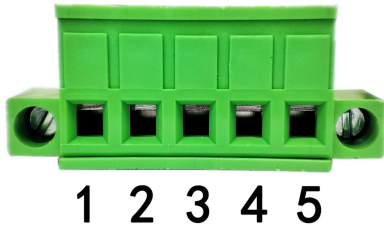
1、以上检测精度均在以下环境条件下测得：温度 25°C、相对湿度 50%RH、1 个大气压，待检测项目最大不超过检测范围。

2、为了持续改进产品，我司保留更改设计功能和规格的权利。

#### 五、产品尺寸



## 六、接线说明



1、+12~24V； 2、GND； 3、RS485A； 4、RS485B； 5、GND1；

电路接线时，把电线与接线端子连接，确保接线正确牢固。

## 七、注意事项

- 1、上电使用需预热 2 分钟以上。
- 2、将模组安装在需要检测的位置，应远离热源。
- 3、请勿将该模组长时间置于高浓度有机气体及腐蚀性气体中。
- 4、请勿将模组安装在强对流气体环境下使用。
- 5、避免地面沙尘，飘絮物等大尘埃颗粒甚至絮状物污染导致风扇缠绕阻转。
- 6、切勿拆解传感器，以防出现不可逆破坏。
- 7、RS485 通讯线需用屏蔽双绞线，必要时需要 2 个 250Ω 终端匹配电阻。
- 8、使用环境应无导电尘埃、无腐蚀金属和破坏绝缘的气体存在，避免直接光照和雨淋。
- 9、避免将模组放在盐雾，酸性或氧化气体（二氧化硫、盐酸等）环境中。



## 八、RS485 通讯协议

### 1、概述

通讯协议详细地描述了环境空气质量检测模组输入和输出命令、信息和数据，以便第三方使用和开发。

本产品采用标准串行 RS485 通讯接口，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/偶检验(通过指令切换)；波特率 9600bps（默认），可通过指令修改；设备地址值出厂默认为 01，可以设置的地址值范围是 01 到 255；（终端电阻可以配置为 120 欧姆，以拨码开关形式切换终端电阻为 120 欧姆）。

### 2、通讯协议详述

#### 2.1 通讯方式

- (1) 所有回路通讯应遵照主/从方式，数据在主站和从站之间传递。
- (2) 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

地址	功能代码	数据数量	数据 1	.....	数据 n	CRCL CRCH
----	------	------	------	-------	------	-----------

### 3、传输格式

#### 3.1 主机发送读地址命令（一对一）：

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	84 27

从机返回地址：

地址	功能码	字节长度	地址高位	地址低位	CRC16 位校验
----	-----	------	------	------	-----------



00	03	02	00	xx	CRCL CRCH
----	----	----	----	----	-----------

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

### 3.2 主机发送读数据命令:

地址	功能码	数据地址高位	数据地址低位	数据个数高位	数据个数低位	CRC 16 位校验
xx	03	00	02	00	07	CRCL CRCH

(目前仅支持同时读取所有数据, 从地址 0x0002 开始, 读取 7 个数据值), 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。

### 从机返回值:

#### 返回格式:

从机地址	功能码	数据个数	数据 N0~N13	CRC 16 位校验
xx	03	0E	xx	CRCL CRCH

#### 返回数据内容:

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

示例:

主机发送: 01 03 00 02 00 07 CRCL CRCH

从机响应: 01 03 0E 02 26 00 26 00 34 00 0E 00 11 00 FF 03 11 CRCL CRCH

#### 注: 数据返回计算

- 1、 CO2 数据: 0x0226, 对应十进制为 550, 表示 CO2 浓度为 550ppm;
- 2、 甲醛数据: 0x0026, 对应十进制为 38, 表示甲醛浓度为 38ppb;



- 3、TVOC 数据: 0x0034, 对应十进制为 52, 表示 TVOC 浓度为 52ppb;
- 4、PM2.5 数据: 0x000E, 对应十进制为 14, 表示 PM2.5 颗粒物浓度为 14ug/m<sup>3</sup>;
- 5、PM10 数据: 0x0011, 对应十进制为 17, 表示 PM10 颗粒物浓度为 17ug/m<sup>3</sup>;
- 6、温度数据: 0x00FF, 对应十进制 255, 表示温度为 25.5°C (负温以补码体现);
- 7、湿度数据: 0x0311, 对应十进制 785, 表示湿度为 78.5%RH;

### 3.3 设置地址命令 (一对一) :

此命令可以设置从机的地址, 地址值 01 到 255 (出厂默认为 01)。

主机发送指令:

地址	功能码	寄存地址高位	寄存器地址低位	从机地址高位	从机地址低位	CRC 16 位校验
00	06	01	00	00	xx	CRCL CRCH

从机应答:

地址	功能码	数据字节数	从机地址高位	从机地址低位	CRC 16 位校验
00	06	02	00	xx	CRCL CRCH

示例:

主机发送: 00 06 01 00 00 address CRCL CRCH

从机应答: 00 06 02 00 address CRCL CRCH

若 address 为 2, 则从机地址设置为 2; 主机得到从机应答后, 设置地址成功。

### 3.4 设置波特率

本产品波特率默认为 9600bps, 可选择更新的波特率 (单位: bps) 有: 4800、9600、11400、19200、38400、57600、115200; 这些波特率分别在指令中对应 1、2、3、4、5、6、7。





设置波特率格式：

地址	功能码	数据地址高位	数据地址低位	数据个数高位	数据个数低位	CRC 16 位校验
xx	06	00	FD	00	xx	CRCL CRCH

示例：

主机发送：01 06 00 FD 00 04 CRCL CRCH

从机响应：无

表示将地址为 01 的设备波特率改为 04，即波特率改为 19200。

### 3.5 无校验和偶校验的切换格式

地址	功能码	数据地址高位	数据地址低位	数据个数高位	数据个数低位	CRC 16 位校验
xx	06	00	FE	00	00/01	CRCL CRCH

00：表示无校验，01 表示偶校验

示例：

主机发送：01 06 00 FE 00 01 CRCL CRCH

从机响应：无

表示将地址为 01 的设备校验方式改为偶校验。

## 4、 主机数据采样频率：

上位机读取数据每次间隔时间不小于 500ms，推荐值 1s 或以上。

## 5、 CRC 校验参考：

```
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-->0) //计算指定长度的 CRC
```

```
{
```

```
    index = crch ^ *ptr++;
```

```
    crch = crcl ^ TabH[index];
```



```
    crcl = TabL[ index];  
  
}  
  
return ((crch<<8) | crcl);  
  
}
```