产品规格书

|  |  |
| --- | --- |
| 产品名称： | 三合一空气质量检测仪 |
| 产品型号： | MAQ020607-S-P6 |
| 版本： | V1.1 |
| 制定人： | GW.H |
| 审核人： | CY |



修改记录表：

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 更改内容 | 更改人 | 更改日期 |
| V1.0 | 新增 | GW.H | 2023.11.20 |
| V1.1 | 优化文字描述、更正PM2.5技术参数 | GW.H | 2023.11.23 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**一、产品描述**

三合一空气质量检测仪是一款数字信号输出的产品，采用RS485输出模式，集激光粉尘PM2.5颗粒物、温度、湿度传感器于一体。可对所处环境进行实时检测，具有良好的稳定性，非常方便客户使用。

**二、应用领域**

* 1. 新风换气系统；
  2. 智能家居设备；
  3. 空气净化器、空调；
  4. 空气质量监测设备；
  5. 厨卫换气控制系统；
  6. 酒店房间空气质量监控。

**三、产品特点**

1. RS485信号输出；
2. 灵敏度高，数据稳定；
3. 温度分辨率0.1℃，湿度分辨率0.1%RH；
4. 最快0.5秒通过RS485信号输出三组传感器监测数据（建议读取数据的时间间隔为1秒）。

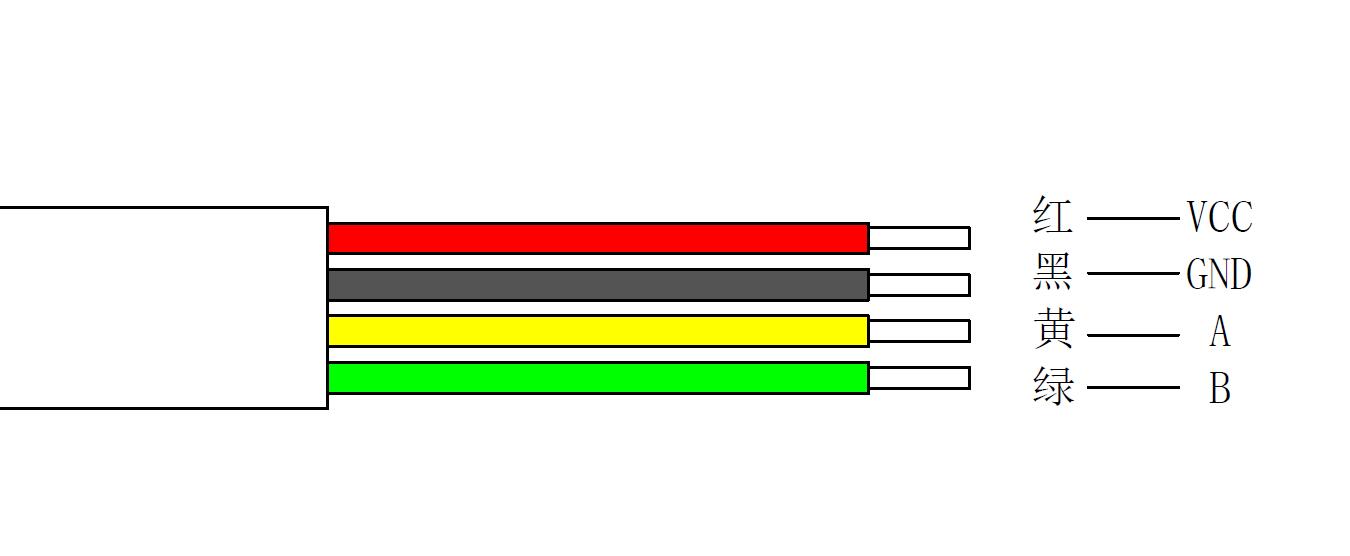
**四、量程精度**

|  |  |  |  |
| --- | --- | --- | --- |
| 类别 | 测量分辨率 | 测量范围 | 测量精度 |
| PM2.5 | 1ug/m3 | 0ug/m³~1000ug/m³ | ±10ug/m³（0~100ug/m³）  ±10%（100~500ug/m³） |
| 温度 | 0.1℃ | -20℃~60℃ | ±1.0℃ |
| 湿度 | 0.1%RH | 5~95%RH | ±5%RH |

**五、技术参数**

|  |  |
| --- | --- |
| 输出信号 | RS485信号 |
| 工作电压 | 12~24VDC |
| 平均工作电流 | ≤20mA |
| 工作温度 | 0~50℃ |
| 工作湿度 | ≤95％RH |
| 外形尺寸 | 直径78mm，高30mm |

六、接线定义

****

**七、注意事项**

1. 本产品适用于普通室内环境；
2. 请勿将本产品安装在强对流气体环境下使用；
3. 用户切勿拆解本产品，以防出现不可逆的破坏；
4. 请勿将本产品长时间置于高浓度有机气体或腐蚀性气体中；
5. 避免地面沙尘、飘絮物等大尘埃颗粒甚至絮状物的污染，导致内部颗粒物传感器的风扇缠绕阻转，建议使用设备采取适当的预过滤处理。

**八、RS485通讯协议**

**1、协议概述**

通信协议详细地描述了本产品的输入和输出命令、信息和数据，以便第三方使用和开发。

物理接口：

1. 连接上位机的主通信口，采用标准串行RS-485通讯口；
2. 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；
3. 数据传输缺省速率为9600b/s。

**2、通信协议详述**

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 | 功能码 | 数据数量 | 数据1 | ... | 数据n | CRC 16位校验 |

**3、传输格式**

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 |

例：发送：00 03 01 00 00 01 84 27；

响应：00 03 02 00 addr CRCL CRCH；若addr=02，则读取的传感器地址为02。

2) 主机发送读全部数据命令：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址 | 功能码 | 数据起始地址高位 | 数据起始地址低位 | 返回数据个数高位 | 返回数据个数低位 | CRC  16位校验 |
| xx | 03 | 00 | 02 | 00 | 07 | xxxx低位在前 |

支持从00 02开始的任意起始地址(起始地址≤00 0E)读取任意个数据（数据个数≤7），下面表格中，未使用的起始地址内的数据为随机数。

内部报文信息：

|  |  |  |
| --- | --- | --- |
| 起始地址 | 字节个数 | 说明 |
| 0x0002 | 2 | 未使用 |
| 0x0004 | 2 | 未使用 |
| 0x0006 | 2 | 未使用 |
| 0x0008 | 2 | PM2.5浓度 |
| 0x000A | 2 | 未使用 |
| 0x000C | 2 | 温度值 |
| 0x000E | 2 | 湿度值 |

从机传感器返回值：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 从机地址 | 功能码 | 字节个数 | 字节N0-N13 | CRC 16位校验 |
| xx | 03 | 0E | …… | xxxx低位在前 |

字节长度仅指数据长度。

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N0 | N1 | N2 | N3 | N4 | N5 | N6 | N7 | N8 | N9 | N10 | N11 | N12 | N13 |
| 未使用 | 未使用 | 未使用 | 未使用 | 未使用 | 未使用 | PM2.5  高位 | PM2.5  低位 | 未使用 | 未使用 | 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；

产品响应：01 03 0E 00 00 00 00 00 00 00 0E 00 11 00 FF 02 58 CRCL CRCH。

产品数据解析：

返回PM2.5数据：0x000E，对应十进制14，表示PM2.5浓度为14 ug/m3；

返回数据：0x0011，这是其它数据，不用理会；

3) 设置地址命令

此命令可以设置从机地址，可修改的地址值范围：1-247，此后，从机地址就是主机设置的地址。

主机发送设置地址命令：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地址 | 功能码 | 寄存器地址高位 | 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 | xxxx低位在前 |

从机返回：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 地址 | 功能码 | 字节长度 | 地址高位 | 地址低位 | CRC 16位校验 |
| 00 | 06 | 02 | 00 | xx | xxxx低位在前 |

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02，则从机地址就被设置为2。

主机得到从机回应后，认为强制设置地址成功，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**

读取本产品传感器数据时，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**

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);

}