

---

# 自动气象站通讯协议

## 目录

一. 协议特性.....	1
二. 协议说明.....	1
1. XPH 通讯协议.....	1
(1).XPH 通用帧格式.....	1
(2).XPH 协议通讯码.....	1
(3).XPH 通讯协议详细说明.....	2
(4).继电器控制相关说明.....	7
2. 标准 MODBUS 通讯协议.....	11
(1).获取 16 通道数据.....	11
(2).获取单独通道数据.....	12
四. 附录.....	12
1.CRC16 校验码的计算.....	12
2.通道数据转换.....	13

## 一. 协议特性

- 1、支持 XPH 通讯、标准 MODBUS 通讯两种格式。
- 2、支持 RS-232、RS-485 通讯总线。
- 3、支持 9600 波特率，特殊版本可达 38400。
- 4、支持最大设备数 254 个。
- 5、数据帧最小间隔 300ms 以上。
- 6、使用 CRC16 数据帧错误检查，多项式码 0xA001。

## 二. 协议说明

自动气象站支持两种格式的通讯协议工作，任一时刻只能使用一种通讯协议进行通讯。下面分别对两种通讯协议进行详细说明。

### 1. XPH 通讯协议

通讯参数：波特率 9600 数据位 8 位 无校验位

XPH 通讯协议是效仿标准 MODBUS 通讯的一个自定义版本，其并不兼容 MODBUS 通讯协议。其特点是召唤式工作，气象站仅作为服务器，使用上位机客户端进行通讯。气象站收到正确数据帧后，就回复相应内容，无效数据帧则不回复任何内容。不回复是便于多台气象站进行 485 总线组网，避免数据通讯冲突。

#### (1).XPH 通用帧格式

序号	内容	字节数	说明	备注
1	地址域	1	气象站通讯地址	唯一
2	读写域	1	读 (0x03) / 写 (0x10) 操作码	2 选 1
3	命令域	2	命令码，前 1 字节必须为 0	00 xx
4	数据长度域	2	数据域长度，不包括本域	发送时可无
5	数据域	N	数据	
6	校验域	2	CRC16 检验码	低前高后

#### (2).XPH 协议通讯码

序号	命令码	含义	类型	备注
1	0x00	实时环境参数数据	通用类型	仅读
2	0x20	系统参数配置	通用类型	读写
3	0x37	读历史数据	通用类型	仅读
4	0x38	历史数据重读	通用类型	仅读
5	0x7A	手动控制命令		仅写
6	0x80	复位气象站	通用类型	仅写



	14	2	通道 6	0x7FFF (无效/未接)
	16	2	通道 7	0x7FFF (无效/未接)
	18	2	通道 8	0x7FFF (无效/未接)
	20	2	通道 9	0x7FFF (无效/未接)
	22	2	通道 10	0x7FFF (无效/未接)
	24	2	通道 11	0x7FFF (无效/未接)
	26	2	通道 12	0x7FFF (无效/未接)
	28	2	通道 13	0x7FFF (无效/未接)
	30	2	通道 14	0x7FFF (无效/未接)
	32	2	通道 15	0x7FFF (无效/未接)
	34	2	通道 16	0x7FFF (无效/未接)
	36	1	继电器 1	01 打开/00 关闭
	37	1	继电器 2	01 打开/00 关闭
	38	1	继电器 3	01 打开/00 关闭
	39	1	继电器 4	01 打开/00 关闭
	40	1	继电器 5	01 打开/00 关闭
	41	1	继电器 6	01 打开/00 关闭
	42	1	继电器 7	01 打开/00 关闭
	43	1	继电器 8	01 打开/00 关闭
	44	1	继电器 9	01 打开/00 关闭
	45	1	继电器 10	01 打开/00 关闭
	46	1	继电器 11	01 打开/00 关闭
	47	1	继电器 12	01 打开/00 关闭
	48	1	继电器 13	01 打开/00 关闭
	49	1	继电器 14	01 打开/00 关闭
	50	1	继电器 15	01 打开/00 关闭
	51	1	继电器 16	01 打开/00 关闭
	52	1	继电器 17	01 打开/00 关闭
	53	1	继电器 18	01 打开/00 关闭
	54	1	继电器 19	01 打开/00 关闭
	55	1	继电器 20	01 打开/00 关闭
	56	1	继电器 21	01 打开/00 关闭
	57	1	继电器 22	01 打开/00 关闭
	58	1	继电器 23	01 打开/00 关闭
	59	1	继电器 24	01 打开/00 关闭
	60	1	继电器 25	01 打开/00 关闭
	61	1	继电器 26	01 打开/00 关闭
	62	1	继电器 27	01 打开/00 关闭
	63	1	继电器 28	01 打开/00 关闭
	64	1	继电器 29	01 打开/00 关闭
	65	1	继电器 30	01 打开/00 关闭
	66	1	继电器 31	01 打开/00 关闭
	67	1	继电器 32	01 打开/00 关闭

5	校验域	68	2	低前高后	0x6B 0xDA
---	-----	----	---	------	-----------

16 通道对应的配置出厂时定义，根据需要而定。

#### 4>. 读取系统参数配置

客户端发送：（6 个字节）

Add 03 00 20 CRC16

例如发送：

01 03 00 20 F0 00

返回：（14 字节）

01 03 00 08 0B 08 1F 0B 00 0A 01 01 98 E0

返回数据说明：

序号	含义	偏移	字节数	说明	备注
1	地址域	0	1	地址(0x01)	0x01
2	操作码	1	1	读(0x03)	0x03
3	数据长度域	2	2	0x00 0x08	高前低后
4	数据域	3	1	年	0B(2011 年)
		4	1	月	08(8 月)
		5	1	日	1F (31 日)
		6	1	时	0B(12 点/24 小时制)
		7	1	分	00(00 分钟)
		8	1	存储间隔	10(分钟)
		9	1	语言类型 0/1	1-英语/0-中文
		10	1	设备地址	01(0x01-0xFE)
5	校验域	11	2	低前高后	0x98 0xE0

存储间隔：即历史数据保存的间隔，单位为：分钟

设备地址：设置范围 0-254，

0 为广播地址，一般不要随便使用，主要用于组网或多播；

1-254 为正常设备地址，同一总线上必须保证唯一；

若无返回数据，则说明读取不成功。

#### 5>. 读取采集仪用户 ID

客户端发送：（8 字节）

Add 03 00 60 00 04 CRC16

例如发送：01 03 00 60 00 04 44 17

返回：（12 字节）

01 03 00 60 00 04 0E 06 0C 01 82 DD

#### 6>. 写采集仪 ID 命令：

客户端发送：（12 字节）

Add 10 00 60 00 04 XX XX XX XX CRC16

例如发送：

01 10 00 60 00 04 0F 09 18 01 65 12

返回：（8 字节）

01 10 00 60 00 04 C1 D4

### 7>. 写入系统参数配置

客户端发送：(17 字节)

Add 10 00 20 00 04 XX XX XX XX XX XX XX XX CRC16

例如发送：

01 10 00 20 00 04 08 0B 08 1F 0B 00 10 00 01 98 1D

返回：(8 个字节)

01 10 00 20 00 04 C0 00

发送数据说明：

序号	含义	偏移	字节数	说明	备注
1	地址域	0	1	地址(0x01)	0x01
2	操作码	1	1	写(0x10)	0x10
3	命令码	2	2	0x00 0x20	高前低后
4	寄存器数量	4	2	0x00 0x04	高前低后
5	数据长度	6	1	0x08	8 个数据字节
6	数据域	7	1	0B	0B(2011 年)
		8	1	08	08(8 月)
		9	1	1F	1F (31 日)
		10	1	0B	0B(12 点/24 小时制)
		11	1	00	00(00 分钟)
		12	1	10	存储间隔 16(分钟)
		13	1	00	1-英语/0-中文
14	1	01	地址		
5	校验域	15	2	低前高后	0x98 0x1D

寄存器数量：该域无意义，保留该域为了向前兼容，请尽量使用说明的数值；

数据域最后一个字节，目前为设备地址设置，之前的通讯版本选择为保留，故在 20110831 之后都要小心设置。最好是先读取系统参数，然后使用读取的设备地址填充该字节。

若无返回数据，则说明写入不成功，否则写入成功。

### 8>. 读取历史数据

客户端发送：(6 个字节)

01 03 00 37 B0 0E

气象站返回：(1044 个字节)

01 03 04 0E 00 1C D0~DN CRC16

返回数据说明：

序号	含义	偏移	字节数	说明	备注
1	地址域	0	1	地址(0x01)	0x01
2	操作码	1	1	读(0x03)	0x03
3	数据长度	2	2	0x04 0x0E	高前低后
4	有效数据帧数	4	2	0x00 0x1C	高前低后
5	数据域	6	1036	历史数据包	分成 N 帧

6	校验域	1042	2	低前高后	CRC16 低前高后
---	-----	------	---	------	------------

数据长度域：为“有效数据帧数”和“数据域”两个域字节总和。

数据域：固定大小为 1036 个字节，每帧 37 字节，故最大有效帧 1036/37=28 帧。

有效数据帧数：由于历史数据包是等长的，该域指示数据域中包含的有效帧数，有效帧从数据域的 0 位置开始，顺序往后排列。最大有效帧为 28 帧。

历史数据帧说明：(37 字节)

序号	偏移	字节数	说明
1	0	1	年(2000 年以后的偏移)
2	1	1	月
3	2	1	日
4	3	1	时(24 小时制)
5	4	1	分钟
16 通道数据值	5	2	通道 1
	7	2	通道 2
	9	2	通道 3
	11	2	通道 4
	13	2	通道 5
	15	2	通道 6
	17	2	通道 7
	19	2	通道 8
	21	2	通道 9
	23	2	通道 10
	25	2	通道 11
	27	2	通道 12
	29	2	通道 13
	31	2	通道 14
	33	2	通道 15
	35	2	通道 16

### 9>. 重读历史数据

客户端发送：

01 03 00 38 F0 0A

返回：

返回数据与“[D]读取历史数据”返回数据格式相同。

说明：

由于远程数据读取时，且历史数据包比较大，可能会发生数据错误。就可能造成读取的历史数据在传输过程中出错，从而导致某些上位机接口程序误判读取历史数据错误。而气象站一旦读取了历史数据后不能重读的话，就会造成某包历史数据丢失的现象。重读历史数据命令用于重复读取上一次读取的历史数据，从而解决错误丢失历史问题。

### 10>. 读取登录密码(该命令需额外支持，默认不支持)

客户端发送：(8 个字节)

01 03 00 61 00 04 15 D7

返回：(12 个字节)

01 03 00 61 00 04 XX XX XX XX CRC16

返回数据说明：

序号	含义	偏移	字节数	说明	备注
1	地址域	0	1	地址(0x01)	0x01
2	操作码	1	1	读(0x03)	0x03
3	命令码	2	2	0x00 0x61	高前低后
4	数据长度域	4	2	0x00 0x04	高前低后
5	数据域	6	4	4 位密码数据	从左到右
6	校验域	10	2	低前高后	CRC16 低前高后

数据域：4 位 ASCII 码表示的密码，每位的范围 0-9，即 0x30-0x39。数据域 0-3 字节对应界面输入的左-右顺序。

#### 11>. 写入登录密码(该命令需额外支持，默认不支持)

客户端发送：(12 字节)

01 10 00 61 00 04 XX XX XX XX CRC16

返回：(8 个字节)

01 10 00 61 00 04 CRC16

写入数据说明：

数据域：4 位 ASCII 码表示的密码，每位的范围 0-9，即 0x30-0x39。数据域 0-3 字节对应界面输入的左-右顺序。

#### 12>. 将气象站通讯协议切换成 Modbus 协议

客户端发送：(4 字节)

00 50 01 8C

返回：(4 个字节)

00 50 01 8C

#### 13>. 复位气象站(气象站会切换成 XPH 协议)

客户端发送：(4 字节)

01 80 01 80

气象站无返回。

### (4).继电器控制相关说明

注：继电器需要硬件支持。

#### 1>. 写继电器控制

发送：

Add 10 00 7A XX(写继电器起始位置，一个字节) XX(写继电器个数，一个字节,01 表示 1 个) XX …(要写的状态，一个字节表示一个继电器的状态，00 表示关 01 表示开，可以一



---

次操作多个继电器) CRC16

**返回:**

Add 10 00 7A XX(写继电器起始位置, 一个字节) XX (要写的状态, 一个字节表示一个继电器的状态, 00 表示关 01 表示开) CRC16

默认地址是 1 , 则控制 16 个继电器发送如下命令:

继电器 1:

打开发送: 01 10 00 7A 00 01 01 D0 18

返回: 01 10 00 7A 00 01 20 10

关闭发送: 01 10 00 7A 00 01 00 11 D8

返回: 01 10 00 7A 00 00 E1 D0

继电器 2:

打开发送: 01 10 00 7A 01 01 01 81 D8

返回: 01 10 00 7A 01 01 21 80

关闭发送: 01 10 00 7A 01 01 00 40 18

返回: 01 10 00 7A 01 00 E0 40

继电器 3:

打开发送: 01 10 00 7A 02 01 01 71 D8

返回: 01 10 00 7A 02 01 21 70

关闭发送: 01 10 00 7A 02 01 00 B0 18

返回: 01 10 00 7A 02 00 E0 B0

继电器 4:

打开发送: 01 10 00 7A 03 01 01 20 18

返回: 01 10 00 7A 03 01 20 E0

关闭发送: 01 10 00 7A 03 01 00 E1 D8

返回: 01 10 00 7A 03 00 E1 20

继电器 5 :

打开发送: 01 10 00 7A 04 01 01 91 D9

返回: 01 10 00 7A 04 01 22 D0

关闭发送: 01 10 00 7A 04 01 00 50 19

返回: 01 10 00 7A 04 00 E3 10

继电器 6 :

打开发送: 01 10 00 7A 05 01 01 C0 19

返回: 01 10 00 7A 05 01 23 40

关闭发送: 01 10 00 7A 05 01 00 01 D9

返回: 01 10 00 7A 05 00 E2 80

继电器 7 :

打开发送: 01 10 00 7A 06 01 01 30 19

返回: 01 10 00 7A 06 01 23 B0

---

关闭发送：01 10 00 7A 06 01 00 F1 D9  
返回：01 10 00 7A 06 00 E2 70

继电器 8：

打开发送：01 10 00 7A 07 01 01 61 D9  
返回：01 10 00 7A 07 01 22 20  
关闭发送：01 10 00 7A 07 01 00 A0 19  
返回：01 10 00 7A 07 00 E3 E0

继电器 9：

打开发送：01 10 00 7A 08 01 01 51 DA  
返回：01 10 00 7A 08 01 27 D0  
关闭发送：01 10 00 7A 08 01 00 90 1A  
返回：01 10 00 7A 08 00 E6 10

继电器 10：

打开发送：01 10 00 7A 09 01 01 00 1A  
返回：01 10 00 7A 09 01 26 40  
关闭发送：01 10 00 7A 09 01 00 C1 DA  
返回：01 10 00 7A 09 00 E7 80

继电器 11：

打开发送：01 10 00 7A 0A 01 01 F0 1A  
返回：01 10 00 7A 0A 01 26 B0  
关闭发送：01 10 00 7A 0A 01 00 31 DA  
返回：01 10 00 7A 0A 00 E7 70

继电器 12：

打开发送：01 10 00 7A 0B 01 01 A1 DA  
返回：01 10 00 7A 0B 01 27 20  
关闭发送：01 10 00 7A 0B 01 00 60 1A  
返回：01 10 00 7A 0B 00 E6 E0

继电器 13：

打开发送：01 10 00 7A 0C 01 01 10 1B  
返回：01 10 00 7A 0C 01 25 10  
关闭发送：01 10 00 7A 0C 01 00 D1 DB  
返回：01 10 00 7A 0C 00 E4 D0

继电器 14：

打开发送：01 10 00 7A 0D 01 01 41 DB  
返回：01 10 00 7A 0D 01 24 80  
关闭发送：01 10 00 7A 0D 01 00 80 1B  
返回：01 10 00 7A 0D 00 E5 40

---

继电器 15 :

打开发送: 01 10 00 7A 0E 01 01 B1 DB

返回: 01 10 00 7A 0E 01 24 70

关闭发送: 01 10 00 7A 0E 01 00 70 1B

返回: 01 10 00 7A 0E 00 E5 B0

继电器 16 :

打开发送: 01 10 00 7A 0F 01 01 E0 1B

返回: 01 10 00 7A 0F 01 25 E0

关闭发送: 01 10 00 7A 0F 01 00 21 DB

返回: 01 10 00 7A 0F 00 E4 20

其它继电器依此类推

## 2>. 写继电器打开持续时间

打开持续时间: 即打开继电器之后持续 XX 秒之后自动关闭, 防止气象站无法通信的时候强电设备持续长时间运行, 默认为 0xFFFF, 即 65535 秒。

**客户端发送:**

Add 10 00 80 XX(继电器起始号, 1 个字节)XX(继电器个数, 1 个字节) XX XX... (数据, 每个继电器 2 个字节) CRC

例如:

设置继电器 1 打开持续时间为 60 秒, 即发送:

01 10 00 80 00 01 00 3C C0 09

说明: 16 进制 0x003C 即 10 进制 60, 即持续时间为 60 秒。

**返回:**

01 10 00 80 01 BD

## 3>. 读继电器闭合持续时间

**客户端发送:**

Add 03 00 80 XX(继电器起始号)XX (继电器个数) CRC16

**返回:**

01 03 00 80 XX (返回字节数) XX XX... (数据, 每个继电器 2 个字节) CRC16

例如:

发送: 01 03 00 80 00 01 85 E2

返回: 01 03 00 80 02 00 3C 83 E2



数据长度 = (结束编号 - 起始编号) \* 2

## (2).获取单独通道数据

序号	地址	字节数	说明
16 通道数据值	40001	2	通道 1 (风速)
	40002	2	通道 2 (雨量)
	40003	2	通道 3 (温度)
	40004	2	通道 4 (气压)
	40005	2	通道 5
	40006	2	通道 6 (辐射)
	40007	2	通道 7 (风向)
	40008	2	通道 8
	40009	2	通道 9 (湿度)
	40010	2	通道 10
	40011	2	通道 11
	40012	2	通道 12
	40013	2	通道 13
	40014	2	通道 14
	40015	2	通道 15
	40016	2	通道 16

注：其他通道未配置读出来为 0X7FFF，数据为高前低后。

如读取通道 3 (温度) 数据：

发送：01 03 00 02 00 01 25 CA

返回：01 03 02 7F FF D8 34

7F FF 表示未接传感器。

## 四. 附录

### 1.CRC16 校验码的计算

- (1). 预置 1 个 16 位的寄存器为十六进制 FFFF (即全为 1)，称此寄存器为 CRC 寄存器；
- (2). 把第一个 8 位二进制数据 (既通讯信息帧的第一个字节) 与 16 位的 CRC 寄存器的低 8 位相异或，把结果放于 CRC 寄存器；
- (3). 把 CRC 寄存器的内容右移一位 (朝低位) 用 0 填补最高位，并检查右移后的移出位；
- (4). 如果移出位为 0：重复第 3 步 (再次右移一位)；如果移出位为 1：CRC 寄存器与多项式 A001 (1010 0000 0000 0001) 进行异或；
- (5). 重复步骤 3 和 4，直到右移 8 次，这样整个 8 位数据全部进行了处理；
- (6). 重复步骤 2 到步骤 5，进行通讯信息帧下一个字节的处理；
- (7). 将该通讯信息帧所有字节按上述步骤计算完成后，得到的 16 位 CRC 寄存器的高、低字节进行交换；

(8). 最后得到的 CRC 寄存器内容即为 CRC16 码。(注意得到的 CRC 码即为低前高后顺序)

附 CRC16 计算代码:

```
u16 CRC16 (u8 * p, u16 datalen)
{
    unsigned char CRC16Lo, CRC16Hi, CL, CH, SaveHi, SaveLo;
    int i, Flag;
    CRC16Lo = 0xFF;
        CRC16Hi = 0xFF;
    CL = 0x01;
        CH = 0xA0;
    for(i=0; i<datalen; i++)
    {
        CRC16Lo ^=*(p+i);           //每一个数据与 CRC 寄存器进行异或
        for(Flag=0; Flag<8; Flag++)
        {
            SaveHi = CRC16Hi;
            SaveLo = CRC16Lo;
            CRC16Hi >>= 1;
            CRC16Lo >>= 1;           //高位右移一位, 低位右移一位
            if ((SaveHi & 0x01) == 0x01) //如果高位字节最后一位为 1
                CRC16Lo |=0x80;       //则低位字节右移后前面补 1 否则自动补 0
            if ((SaveLo & 0x01) == 0x01) //如果 LSB 为 1, 则与多项式码进行异或
            {
                CRC16Hi ^= CH;
                CRC16Lo ^= CL;
            }
        }
    }
    return (CRC16Hi<<8)|CRC16Lo;
}
```

## 2.通道数据转换

例如温度十六进制编码为“00 C3”，换算成二进制为“0000000 11000011”，其二进制的第一位为“0”，所以它的值为正数，换算出十进制值“195”，最后将它除以 10 后得到最终结果“19.5”。

又例如温度十六进制编码为“FF 3D”，换算成二进制为“11111111 00111101”，其二进制的第一位为“1”，所以它的值为负数。

其具体换算步骤如下:

- (1) 将其二进制的的第一位替换为“0” 得到: “01111111 00111101”
- (2) 后 15 位取反后得到: “00000000 11000010”
- (3) 加上“1” 后得到: “00000000 11000011”
- (4) 按照 1 中的正数表示方法得到十进制值“195”
- (5) 因为是负值所以为“-195”

---

(6) 结果除以 10, 最终结果为 “-19.5”

所以: 00 C3 → 19.5°C

FF 3D → -19.5°C