**新普惠自动气象站通讯协议**

V20160307

目录

[一．协议特性 2](#_Toc445193068)

[二．协议说明 2](#_Toc445193069)

[1．XPH通讯协议 2](#_Toc445193070)

[(1).XPH通用帧格式 2](#_Toc445193071)

[(2).XPH协议通讯码 2](#_Toc445193072)

[(3).XPH通讯协议详细说明 3](#_Toc445193073)

[(4).继电器控制相关说明 9](#_Toc445193074)

[2．标准MODBUS通讯协议 10](#_Toc445193075)

[(1).获取16通道数据 11](#_Toc445193076)

[(2).获取单独通道数据 11](#_Toc445193077)

[四．附录 12](#_Toc445193078)

[1.CRC16校验码的计算 12](#_Toc445193079)

[2.通道数据转换 13](#_Toc445193080)

# 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

新普惠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，就回复相应的内容，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 |

(3).XPH通讯协议详细说明

对于每条通讯命令码，进行收发过程的详细描述，假定气象站的地址为0x01。返回的通道数据都以有符号16进制数表示，如通道1的数据（包括实时数和历史数据）为0x12 0x34，表示为十进制数即4660(0x1234)。即数据字节的高位在前，低位在后，此顺序不能错，否则数据值将错误。

另外，对不不同的传感器类型，某些通道数据值需要表示N位小数，此时，该通道16位有符号数扩大10的N次方倍。如通道1表示风速，带一位小数，其数据为0x00 0x7B(十进制123)，则实际风速为12.3m/s，即读取的数据扩大了10倍，其他小数以此类推。传感器精度参数详见说明书。

**1>.读气象站地址命令：**

客户端发送：（4字节）

00 20 00 68 00为广播地址，在地址未知的情况下也可以通讯

返回：（5字节）

00 20 Add CRC16

**2>.写气象站地址命令:**

客户端发送：（7字节）

00 10 00 81 Add CRC16

例如发送：

00 10 00 81 01 81 50

返回：（5字节）

00 10 01 BD C0

**3>.读取实时环境数据**

客户端发送：（6字节）

Add 03 00 00 CRC16

例如发送：

01 03 00 00 F1 D8

返回：(70字节)

01 03 00 40 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 6B DA

返回数据格式说明：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 | 仅读(0x03) | 0x03 |
| 3 | 数据长度域 | 2 | 2 | 0x00 0x40 | 高前低后 |
| 4 | 数据域 | 4 | 2 | 通道1 | 0x7FFF(无效/未接) |
| 6 | 2 | 通道2 | 0x7FFF(无效/未接) |
| 8 | 2 | 通道3 | 0x7FFF(无效/未接) |
| 10 | 2 | 通道4 | 0x7FFF(无效/未接) |
| 12 | 2 | 通道5 | 0x7FFF(无效/未接) |
| 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) |
| 5 | 继电器控制 | 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 |
| 6 | 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

返回:Add 03 00 60 xx xx xx xx 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

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 0x60 | 高前低后 |
| 4 | 数据长度 | 4 | 2 | 0x00 0x04 | 表示返回的有效数据是4个字节 |
| 5 | 设备编号 | 6 | 4 | 0x0E 0x06  0x0C 0x01 | 即十进制14 06 12 01,拼接而成就是14061201 |
| 6 | CRC16 | 10 | 2 | 0x82 0xDD | 低前高后 |

**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

返回:Add 10 00 60 00 04 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

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 0x60 | 高前低后 |
| 4 | 数据长度 | 4 | 2 | 0x00 0x04 | 表示要发送4个字节的数据 |
| 5 | 要写入的设备编号 | 6 | 4 | 0x0F 0x09  0x18 0x01 | 即10进制15 09 24 01,即写入的设备编号为15092401 |
| 6 | CRC16 | 10 | 2 | 0x65 0x12 | 低前高后 |

**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 | 0x98 0x1D | 低前高后 |

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.

**14>.读取电源信息/GPRS信号**

发送:Add 03 00 21 CRC16

回复:Add 03 00 06 xx xx xx xx xx xx CRC16

例如：01 03 00 21 31 C0

回复：01 03 00 06 00 00 00 00 00 00 53 32

第5位开始00 00 00 00 00 00 依次为电源电压,电量百分比,GPRS信号强度，都是两个字节高前低后,电压除以1000,即保留3位小数,以V为单位,例如12.353V,电量百分比为整数例如95%,GPRS信号为整数.

**15>.切换电源工作模式**

**切换省电模式**

发送:Add 10 00 23 00 CRC16

回复:Add 10 00 23 00 CRC16

00表示省电模式

**切换常规模式**

发送:Add 10 00 23 01 CRC16

回复:Add 10 00 23 01 CRC16

01表示非省电模式

(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(继电器编号，一个字节) XX（写继电器状态, 一个字节,01开/00关）CRC16

**返回：**

Add 10 00 7A XX(继电器编号,一个字节) XX（继电器开关状态 01开, 00关） CRC16

继电器编号为0~31,表示第1~32路继电器.

**默认地址是1则控制32个继电器发送如下命令：**

**继电器0(即第一个继电器)：**

打开发送：01 10 00 7A 00 01 CRC16

返回：01 10 00 7A 00 01 20 10

关闭发送：01 10 00 7A 00 00 CRC16

返回：01 10 00 7A 00 00 E1 D0

**继电器1：**

打开发送：01 10 00 7A 01 01 CRC16

返回：01 10 00 7A 01 01 21 80

关闭发送：01 10 00 7A 01 00 CRC16

返回：01 10 00 7A 01 00 E0 40

**其他继电器依此类推.**

**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

2．标准MODBUS通讯协议

通讯参数：波特率9600 数据位 8位 无校验位

气象站标准MODBUS通讯协议使用RTU 模式。消息帧发送至少要以3.5 个字符时间的停顿间隔开始，即在传输完每帧的最后一个字节，至少需要3.5 个字符时间的停顿来标定消息帧的结束。下一个新的消息帧可在此停顿后开始，整个消息帧必须作为一连续的流传输。

气象站标准MODBUS通讯协议仅支持读功能，不支持配置气象站寄存器的功能，即不能写入任何数据。其特点有：

①气象站共有16个通道寄存器，对应16个通道的传感器测量输出值；

②寄存器最小编号固定于0，最大编号固定于15，不可更改；

③支持一次读取单个、多个（寄存器编号必须连续）寄存器；

④每一通道数据占2个字节，高字节在前，低字节在后，16位有符号整形值；

⑤数据小数位定义，同XPH协议的小数定义，即N位小数，数据值扩大10的N次方倍。

**气象站标准MODBUS通讯协议**

(1).获取16通道数据

发送：Add 03 00 00 00 10 CRC16

例如： 01 03 00 00 00 10 44 06

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 含义 | 偏移 | 字节数 | 说明 |
| 1 | 设备地址 | 0 | 1 | 设备唯一标识 |
| 2 | 操作码(读) | 1 | 1 | 固定值0x03 |
| 3 | 寄存器起始编号 | 2 | 2 | 读取的第一个寄存器编号 |
| 4 | 寄存器个数 | 4 | 2 | 读取寄存器个数 |
| 5 | CRC16检验 | 6 | 2 | 低前高后 |

设备地址：假定为0x01，有效范围0-254，0为广播地址。

操 作 码：固定为0x03，即读取操作，不支持其他操作。

起始编号：范围0-15，表示要读取的第一个寄存器编号。

寄存器个数：范围1-16

气象站返回数据帧：

如 01 03 20 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 7F FF 8C 45

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 含义 | 偏移 | 字节数 | 说明 |
| 1 | 设备地址 | 0 | 1 | 设备唯一标识 |
| 2 | 操作码(读) | 1 | 1 | 固定值0x03 |
| 3 | 数据长度 | 2 | 1 | 不含本字节 |
| 4 | 数据域 | 3 | N | 最多32个字节 |
| 5 | CRC16检验 | 3+N | 2 | 低前高后 |

数据长度：不含本身，仅表示数据域的字节数量。最大32，最小0。

根据读取指令序列的“起始编号”和“结束编号”来确定。

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℃

FF 3D → -19.5℃