Modbus 通讯协议使用说明

索引

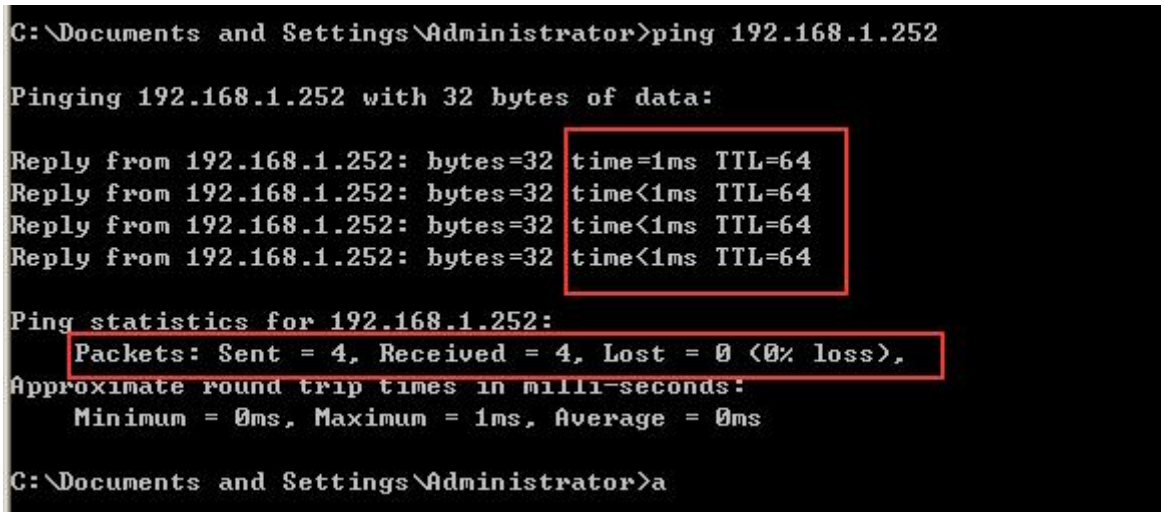
1、通讯配置

通信接口：标准 2 线 RS485 或标准 3 线 RS232，光电隔离，ESD 保护
波特率 ：2400 ~ 115200， 通过仪表界面中设置
通讯地址：1~99
通讯协议：Modbus/Modbus-2
通信格式：8 位数据位，无校验位，1 位停止位（8/N/1）

提示：控制器主界面屏左上，右上空白处点一下，提示输入系统密码时输入 0 进入系统设定（提示：串口通讯，网络设定两项在同一页），进入串口通讯

网络设定:
仪表 IP:192.168.1.252（根据需要可设成其它，但控制器 IP 必须和电脑 IP 在同一网段）
网络通讯开关：ON
网络通讯端口号：50000（默认 50000，根据需要进行更改，取值范围 1－50000）
*设置好后控制器断电重启。

PC 端 win/command+R 键输入 cmd 回车，出现如下窗口，输入 ping 192.168.1.252



*类似这样就代表电脑和仪表的网络连接通畅。

2、通讯指令和数据格式

1. 机器可以提供连续读寄存器、写单个寄存器、连续写多个寄存器 MODBUSRTU 通讯指令；（最大连续读/写 100 个寄存器）
2. 功能码。
   03H:为读连续多个 WORD 寄存器的功能码
   06H:更改一个 WORD 寄存器的功能码
   10H:更改写连续多个 WORD 寄存器的功能码
3. 例子:

<1>功能码 03H 的例子(16 进制)

要读取 6 个通道的测量值(PV):

发送: 01 03 00 02 00 06 64 08

01：地址 要询问的设备号地址

03：功能码 表示连续读一段寄存器

00 02：读取的开始位置

00 06：读取的个数

64 08：CRC 校验

下位机回发: 01 03 0C 00 64 00 E6 00 FA 01 5E 01 C2 02 30 C9 42

模块返回第 1~6 通道的测温值（PV）依次为：

01：地址 表示这个报文是 1 号设备发过来的

03：功能码 表示连续读一段寄存器

0C：返回的数据长度 12 个字节

00 64: 第 1 通道（10.0℃）；

00 E6: 第 2 通道（23.0℃）；

00 FA: 第 3 通道（25.0℃）；

01 5E: 第 4 通道（35.0℃）；

01 C2: 第 5 通道（45.0℃）；

02 30: 第 6 通道（56.0℃）；

C9 42: CRC 校验

<2> 功能码 06 写单个寄存器 (16 进制)

发: 01 06 00 65 21 98 80 2F

收: 01 06 00 65 21 98 80 2F 正确返回一样的报文

01：地址 表示这个报文是 1 号设备发过来的

06：功能码 表示要修改一个寄存器

00 65 ：要修改的目标地址

21 98 ：修改的值

80 2F：CRC 校验

<3> 功能码 10 写多个连续寄存器(16 进制)

发:01 10 00 65 00 02 04 0E 10 00 FD F7 14

收:01 10 00 65 00 02 51 D7

01：地址 表示这个报文是 1 号设备发过来的

10: 功能码 写多个连续寄存器

00 65: 起始地址

00 02：连续要写的个数

04：数据长度

0E 10：第 1 个地址要修改的内容

00 FD: 第 2 个地址要修改的内容

F7 14: CRC 校验

3、数据发送和接收格式说明

MODBUS-RTU 报文模式

控制器地址	功能码	数据格式	CRC 校验 L	CRC 检验 H
8bit	8bit	N*8bit	8bit	8bit

PC 对控制器写操作

0x01	06	00 01	00 17	98 04
从机地址	功能码	数据地址	数据	CRC 校验

PC 对控制器读操作

0x01	03	00 01	00 01	D5 CA
从机地址	功能码	数据地址	数据个数	CRC 校验

控制器对 PC 返回内容

0x01	03	01	00 17	F8 4A
从机地址	功能码	数据字节个数	两个字节数据	CRC 校验

4、Modbus-RTU 与 Modbus-TCP 区别

Modbus-RUT 定值启动指令						01 06 00 01 00 03 CC CC							
Modbus-TCP 定值启动指令						00 00 00 00 00 06 01 06 00 01 00 03							
												校验码	
						01	06	00	01	00	03	98	0B
00	00	00	00	00	06	01	06	00	01	00	03		
事务处理标识符（顺序号）		协议标识符（00 00）		随后字节数量		串口和网络这部分一样							

*Modbus-RTU 通讯需要校验码，Modbus-TCP 通讯不需要校验码

读不连续寄存器地址，最大读 200 个寄存器

例：读 3 号，7 号寄存器，返回每个值 4 个字节

01 47 00 03 00 07 04 07

写不连续寄存器地址，最大写 200 个寄存器

例：写 3 号，7 号寄存器，每个值 4 个字节

01 48 00 03 00 00 13 65 00 07 00 00 08 19 42 D9

5、常用寄存器地址说明解析

例：从 0 号寄存器连续读 6 个值

PC 发送指令：01 03 00 00 00 06 C5 C8

PC 接收报文：01 03 0C 08 36 00 02 13 F0 00 50 22 60 03 02 70 0A

00 02 定值停止，00 03 是定值启动, 00 01 程式启动，00 00 程式停止。

6、程式相关寄存器地址说明解析

03 Read Holding Registers (4x).....	Data Addresses	Parameter name	读写	倍数	备注
40002[1]	[0x01]	启动/停止（读/写）	RW	1	01 06 00 00 02 01 cc cc //0 号地址 2 号位置 1 0bit: 1=运行中 0=停止中 1bit: 1=定值 0=程式 2bit: 1=暂停中 0=非暂停中 3bit: 1=保持中 0=非保持中 4bit: 1=跳段中 5bit: 1=待机中 6bit: 1=发生故障 7bit: 1=运行结束 报文举例： 定值启动 01 06 00 01 **00 03** 98 0B 定值停止 01 06 00 01 **00 02** 59 CB 程式启动 01 06 00 01 **00 01** 19 CA 程式停止 01 06 00 01 **00 00** D8 0A 启动定值暂停: 01 06 00 01 **00 07** 99 C8 退出定值暂停: 01 06 00 01 **00 03** 98 0B 启动程式暂停: 01 06 00 01 **00 05** 18 09 退出程式暂停: 01 06 00 01 **00 01** 19 CA 启动程式保持: 01 06 00 01 **00 09** 18 0C 退出程式保持: 01 06 00 01 **00 01** 19 CA 启动程式跳段: 01 06 00 01 **00 11** 18 06
40003[2]	[0x02]	温度测量值	RO	0.01	
40004[3]	[0x03]	湿度测量值	RO	0.1	
40005[4]	[0x04]	温度设定值	RW	0.01	在斜率作用下运行时会发生变化当前设定值
40006[5]	[0x05]	湿度设定值	RW	0.1	
40007[6]	[0x06]	温度出力	RO	0.1	
40008[7]	[0x07]	湿度出力	RO	0.1	
40009[8]	[0x08]	运行时间（小时）	RO	1	
40010[9]	[0x09]	运行时间（秒）	RO	1	
40011[10]	[0x0A]	剩余时间（小时）	RO	1	
40012[11]	[0x0B]	剩余时间（秒）	RO	1	
40013[12]	[0x0C]	定值运行时间（小时）	RW	1	
40014[13]	[0x0D]	定值运行时间（秒）	RW	1	
40015[14]	[0x0E]	需要执行的程式号	RW	1	
40016[15]	[0x0F]	当前程式段号	RO	1	例：601 表示共有 6 段，当前在第 1 段
40017[16]	[0x10]	已循环次数	RO	1	
40018[17]	[0x11]	总循环次数目标值	RO	1	
40019[18]	[0x12]	部分循环次数	RO	1	
40020[19]	[0x13]	部分循环次数设定	RO	1	
40021[20]	[0x14]	WATCH_STATUS1	RO	1	状态灯（0-无动作 1-动作） 0bit:第 1 号的状态灯 1bit:第 2 号的状态灯 ....
40022[21]	[0x15]	WATCH_STATUS2	RO	1	预留

03 Read Holding Registers (4x).....	Data Addresses	Parameter name	读写	倍数	备注
40023[22]	[0x16]	DI_STATUS1 报警	RO	1	报警状态（0－无报警，1－有报警） 0bit:DI1 的状态 1bit:DI2 的状态 ....
40024[23]	[0x17]	DI_STATUS2 报警	RO	1	预留
40025[24]	[0x18]	DO_STATUS1	RO	1	0bit:继电器 1 的状态 1bit:继电器 2 的状态 ...
40026[25]	[0x19]	DO_STATUS2	RO	1	预留
40027[26]	[0x1A]	TS1	RO	1	取个位表示当前 TS 动作状态
40028[27]	[0x1B]	TS2	RO	1	
40029[28]	[0x1C]	TS3	RO	1	
40030[29]	[0x1D]	TS4	RO	1	
40031[30]	[0x1E]	温度设定值	RW	0.01	目标设定值 在斜率作用下运行时不会发生变化
40032[31]	[0x1F]	湿度设定值	RW	0.1	
40033[32]	[0x20]	温度斜率	RW	0.01	
40034[33]	[0x21]	湿度斜率	RW	0.1	
40035[34]	[0x22]	CH1_STATUS	RO	1	通道状态 0bit: 1=运行中 0=停止中 1bit: 1=传感器异常 2bit: 1=演算中 3bit: 1=PV 到达了 4bit: 1=上升中 5bit: 1=保持中 6bit: 1=下降中 7bit: 1=斜率中 9bit：0＝有 SV 1＝无 SV
40036[35]	[0x23]	CH2_STATUS	RO	1	
40037[36]	[0x24]	CH1_KP	RO	1	第 1 通道 PID 的 KP
40038[37]	[0x25]	CH2_KP	RO	1	第 2 通道 PID 的 KP
40039[38]	[0x26]	CH1_TI	RO	1	第 1 通道 PID 的 TI
40040[39]	[0x27]	CH2_TI	RO	1	第 2 通道 PID 的 TI
40041[40]	[0x28]	CH1_TD	RO	1	第 1 通道 PID 的 TD
40042[41]	[0x29]	CH2_TD	RO	1	第 2 通道 PID 的 TD
40043[42]	[0x2A]	CH1_SV_LOW	RO	0.01	CH1 设定值低限
40044[43]	[0x2B]	CH2_SV_LOW	RO	0.1	CH2 设定值低限
40045[44]	[0x2C]	CH1_SV_HIGH	RO	0.01	CH1 设定值高限
40046[45]	[0x2D]	CH2_SV_HIGH	RO	0.1	CH2 设定值高限
41001[1000]	[0x03E8]	运行方式	RW	1	0－程式，100－定值
41002[1001]	[0x03E9]	程式运行下要执行的程式号	RW	0.01	
41003[1002]	[0x03EA]	定值定时运行时间	RW	1	例：306＝3 小时 06 分钟
41005[1004]	[0x03EC]	背光时间	RW	0.01	
41016[1015]	[0x03F7]	掉电模式	RW	1	0－停止，100－冷启，200－热启
41020[1019]	[0x03FB]	待机开关	RW	1	100＝打开，0=关闭
41021[1020]	[0x03FC]	待机时间	RW	1	例：130＝1 小时 30 分钟

03 Read Holding Registers (4x).....	Data Addresses	Parameter name	读写	倍数	备注
41405[1404]	[0x057C]	温度待机区域	RW	0.01	
41465[1464]	[0x05B8]	湿度待机区域	RW	0.01	
42001[2000]	[0x07D0]	当前编辑程式号	RW	1	有＝编辑设定程式号，无＝创建设定程式号
42002[2001]	[0x07D1]	程式循环次数	RW	1	
42003[2002]	[0x07D2]	连接的程式号	RW	1	当前程式组结束后需要继续执行的程式号 注意：写入值为 4444 时表示删除这个程式，这个程式在运行中不可删除
42004[2003]	[0x07D3]		RW		程式名称共 18 个字节，一个地址存 2 个字节，低位在前高位在后。
42005[2004]	[0x07D4]		RW		
	...		RW		
42013[2012]	[0x07DC]	当前程式段数	RO	1	
42014[2013]	[0x07DD]	段循环 1 开始段	RW	1	
42015[2014]	[0x07DE]	段循环 1 结束段	RW	1	
42016[2015]	[0x07DF]	段循环 1 循环数	RW	1	
42017[2016]	[0x07E0]	段循环 2 开始段	RW	1	
42018[2017]	[0x07E1]	段循环 2 结束段	RW	1	
42019[2018]	[0x07E2]	段循环 2 循环数	RW	1	
42020[2019]	[0x07E3]	段循环 3 开始段	RW	1	
42021[2020]	[0x07E4]	段循环 3 结束段	RW	1	
42022[2021]	[0x07E5]	段循环 3 循环数	RW	1	
42023[2022]	[0x07E6]	段循环 4 开始段	RW	1	
42024[2023]	[0x07E7]	段循环 4 结束段	RW	1	
42025[2024]	[0x07E8]	段循环 4 循环数	RW	1	
42026[2025]	[0x07E9]	程式第 1 段温度	RW	0.01	
42027[2026]	[0x07EA]	程式第 1 段湿度	RW	0.1	
42028[2027]	[0x07EB]	程式第 1 段时间	RW	1	100＝1 小时，155＝1 小时 55 分钟 写入 4444 时表示删除此段(注 意), 写入 8888 时表示插入此段(注 意)
42029[2028]	[0x07EC]	程式第 1 段 TS1	RW	1	TS：（0～7） 0=OFF 1=ON 2=周期 1 3.=周期 2.....
42030[2029]	[0x07ED]	程式第 1 段 TS2	RW	1	
42031[2030]	[0x07EE]	程式第 1 段 TS3	RW	1	
42032[2031]	[0x07EF]	程式第 1 段 TS4	RW	1	
42033[2032]	[0x07F0]	程式第 2 段温度	RW	0.01	
	...	...	...		以此类推，不再赘述，最大 100 段。。。

7、CRC 的相关代码如下

```
unsigned short Cal_CRC_Code(const unsigned char *ucpdata,int len)
{
    unsigned short crc=0xffff;
    unsigned char temp;
    int n;
    while(len--)
    {
        crc=*ucpdata^crc;
        for(n=0;n<8;n++){
            char TT;
            TT=crc&1;                   //检查最低位是否为 1
            crc=crc>>1;                 //crc 寄存器内容右移一位>>1
            crc=crc&0x7fff;             //crc 寄存器最高位补 0,不改变其他位
            if (TT==1){                 //检查最低位是否为 1
                crc=crc^0xa001;         //与多项式 1010 0000 0000 0001 相异或
                crc=crc&0xffff;         //crc 码
            }
        }
                ucpdata++;
    }
    return crc;
}
***
```