

MODBUS RTU 通信协议

本通信协议采用标准 ModBus 协议，采用 RTU（十六进制数）传输模式。ModBus 协议是一种主---从式协议。任何时刻只有一个设备能够在线路上进行发送。由主站管理信息交换，且只有主站能发起。主站会依次对从站进行轮流查询。只有当从站地址与轮询地址相匹配，从站才能回复消息。从站之间不能进行直接通信。协议帧中不包含任何消息报头及消息结束符，消息的开始和结束依靠间隔时间来识别，当间隔时间长于或等于 3.5 个字符时，即作为检测到帧结束。如果网络内没有与查询地址相一致的从站或从站接收时 CRC 校验出错，主站将不会接收到返回帧，这时主站根据超时设定判断是否超时，如超时，作出重发或弹出异常错误窗口动作。

协议帧定义如下：

从站地址	功能代码	数据区	CRC16
------	------	-----	-------

从站地址：地址必须在 1---253（01H~FDH）之间。

在同个主站网络中每个从站地址必须唯一。

00 地址为广播应答地址，255(FFH)为广播不应答地址

功能代码：包含读、写多个寄存器。

数据：以二进制代码传输。

CRC16：循环冗余校验，校验从从站地址到数据区最后一个字节，计算多项式码为 A001(hex)。

通讯口设置

通讯方式 异步串行通讯接口，如 RS-485，RS-232，M-bus 等

波特率 300~38400bps（可由设定仪表二级参数自由更改，设定仪表二级参数 BT，默认 9600）见表※2

字节数据格式 HEX

- 一位起始位
- 八位数据位
- 一位停止位
- 偶校验



消息帧格式 （读、写功能是从主站角度定义的）

读寄存器帧

从站地址	功能代码	首寄存器地址	寄存器数 N	CRC16
1 字节	1 字节	2 字节	2 字节	2 字节
1---253	03H	AddrH, AddrL	NH, NL (1---120)	CrcL, CrcH

读寄存器返回帧

从站地址	功能代码	字节数	寄存器数据	CRC16
1 字节	1 字节	1 字节	N*2 字节	2 字节
1---253	03H	N*2	DataH, DataL	CrcL, CrcH

写寄存器帧

从站地址	功能代码	首寄存器地址	寄存器数 N	字节数	寄存器数据	CRC16
1 字节	1 字节	2 字节	2 字节	1 字节	N*2 字节	2 字节
1---253	06H	AddrH, AddrL	NH, NL 1---120	N*2	DataH, DataL	CrcL, CrcH

写寄存器返回帧

从站地址	功能代码	首寄存器地址	寄存器数 N	CRC16
1 字节	1 字节	2 字节	2 字节	2 字节
1---253	06H	AddrH, AddrL	NH, NL 1---120	CrcL, CrcH

错误返回帧

从站地址	功能代码	错误代码	CRC16
1 字节	1 字节	1 字节	2 字节
1---253	功能代码+80H	见表 2	CrcL, CrcH

功能代码表：1

功能代码	ModBus 名	功能名	广播	一次连续的 N 的最大值
03H	Read Holding Registers	读 N 个寄存器值	yes	120
06H	Write Multiple Registers	写 N 个寄存器值	yes	120

错误代码表：2

03 命令错误代码		06 命令错误代码	
代码	说明	代码	说明
1	功能码错误	01	功能码错误
2	寄存器起始地址+寄存器数量错误	02	寄存器地址错误
3	寄存器数量错误	03	寄存器值错误
4	读寄存器失败	04	写单个寄存器失败

单路显示仪表的保持寄存器

03, 06 命令对应的保持寄存器地址表

序号	寄存器地址 (DEC)	参 数 名 称	数据格式	类型	备注
1	00	设备地址	Char	读写	※1 HEX 码
2	01~07	设备序列号	Char	只读	HEX 码
3	08	软件版本号	Char	只读	HEX 码
4	09	硬件版本号	Char	只读	HEX 码
5	0A	通信速率	Char	读写	※2 HEX 码
6	19	工作模式	Char	读写	※3 HEX 码
7	74	阀门开度	Char	读写	00H-64H (即开度 0%-100%) HEX 码
8	90	状态寄存器	Char	只读	※4 HEX 码
以下为带 PID 算法的寄存器地址表内容					
9	91	状态寄存器	Char	只读	※5 HEX 码
10	92	状态寄存器	Char	只读	※6 HEX 码
11	93	状态寄存器	Char	只读	※7 HEX 码
12	A2、A3	第 1 路实时测量值	Float	只读	A2 高字节、A3 低字节, 压缩 BCD 码
13	A4、A5	第 2 路实时测量值	Float	只读	A4 高字节、A5 低字节, 压缩 BCD 码
14	B2、B3	设置目标值	Float	读写	B2 整数、B3 小数, 压缩 BCD 码
15	BC、BD	第 1 路满量程值	Float	读写	BC 整数、BD 小数, 压缩 BCD 码
16	BE、BF	第 2 路满量程值	Float	读写	BE 整数、BF 小数, 压缩 BCD 码
17	C0、C1	第 1 路高报警值	Float	读写	C0 整数、C1 小数, 压缩 BCD 码
18	C2、C3	第 2 路高报警值	Float	读写	C2 整数、C3 小数, 压缩 BCD 码
19	C4、C5	第 1 路低报警值	Float	读写	C4 整数、C5 小数, 压缩 BCD 码
20	C6、C7	第 2 路低报警值	Float	读写	C6 整数、C7 小数, 压缩 BCD 码
21	D0	第 1 路高报警开度	Char	读写	HEX 码 FFH 无效
22	D1	第 2 路高报警开度	Char	读写	
23	D2	第 1 路低报警开度	Char	读写	
24	D3	第 2 路低报警开度	Char	读写	
25	D4、D5	第 1 路高报警比例	Float	读写	前者整数位, 后者小数位 BCD 码 2 字节同时=00H 任一字节=FFH 无效
26	D6、D7	第二路高报警比例	Float	读写	
27	D8、D9	第 1 路低报警比例	Float	读写	
28	DA、DB	第二路低报警比例	Float	读写	
29	DC、DD	正常调节比例	Float	读写	压缩 BCD 码 00H 或非时钟数制无效, 例如 00 时 00 分 00 秒或 27 时 81 分 62 秒 无效
30	E0、E1、E2	采样调整周期	Char	读写	
31	E3、E4、E5	积分时间	Char	读写	
32	E6、E7、E8	微分时间	Char	读写	

※1

修改阀门地址 请求 PDU

地址	功能码	寄存器地址		寄存器值		校验 CRC	
1 字节	1 字节	高位	低位	高位	低位	低位 CRCL	高位 CRCH
00H	06H	00H	01H	00H	01H~F7H	1 字节	1 字节

修改阀门地址 应答 PDU

地址	功能码	寄存器地址		寄存器值		校验 CRC	
1 字节	1 字节	高位	低位	高位	低位	低位 CRCL	高位 CRCH
01H~F7H	06	00H	01H	00H	01H~F7H	1 字节	1 字节

※2

修改阀门通信速率 请求 PDU

地址	功能码	寄存器地址		寄存器值		校验 CRC	
1 字节	1 字节	高位	低位	高位	低位	低位 CRCL	高位 CRCH
00H	06H	00H	0AH	00H	01H~08H	1 字节	1 字节

不应答

修改波特率后，应使用新的波特率通信

波特率 寄存器值对应表

波特率	寄存器值	波特率	寄存器值
300	1	4800	5
600	2	9600	6 或其他值
1200	3	19200	7
2400	4	38400	8

出厂默认波特率为 9600

※3

19H	Bit7~bit1	Bit0	
	保留	=0 命令控制	=1 自动控制

※4

Bit 0 关到位开关失效 =0 有效 =1 失效
Bit 1 开到位开关失效 =0 有效 =1 失效
Bit 2
Bit 3 =00 电机未运行 01 正在开阀, 10 正在关阀
Bit 4 开到位开关状态 =0 开到 100 =1 未开到 100
Bit 5 关到位开关状态 =0 关到 0 =1 未关到 0
Bit 6 阀门开关方向 =0 开放向 =1 关方向
Bit 7 阀门卡死 =0 正常 =1 卡死

※5

Bit 0 采样时间 =0 有效 =1 无效
Bit 1 积分时间 =0 有效 =1 无效
Bit 2 微分时间 =0 有效 =1 无效
Bit 3 调节比例 =0 有效 =1 无效
Bit 4 目标压差 =0 有效 =1 无效
Bit 5 间隙 =0 有效 =1 无效
Bit 6 死区 =0 有效 =1 无效

※6

Bit 0 第 1 路量程值 =0 有效 =1 无效
Bit 2 第 2 路量程值 =0 有效 =1 无效
Bit 3 第 1 路 满量程值 AD 值 =0 有效 =1 无效
Bit 4 第 1 路 0 刻度值 AD 值 =0 有效 =1 无效
Bit 5 第 2 路 满量程值 AD 值 =0 有效 =1 无效
Bit 6 第 2 路 0 刻度值 AD 值 =0 有效 =1 无效

※7

Bit0 第 1 路高报警对应压差值 =0 有效 =1 无效
Bit1 第 2 路高报警对应压差值 =0 有效 =1 无效
Bit2 第 1 路低报警对应压差值 =0 有效 =1 无效
Bit3 第 2 路低报警对应压差值 =0 有效 =1 无效
Bit4 第 1 路高报警对应比例 =0 有效 =1 无效
Bit5 第 2 路高报警对应比例 =0 有效 =1 无效
Bit6 第 1 路低报警对应比例 =0 有效 =1 无效
Bit7 第 2 路低报警对应比例 =0 有效 =1 无效

CRC 校验说明

CRC 校验 2 字节

循环冗余校验 (CRC) 域为两个字节, 包含一个二进制 16 位值。附加在报文后面的 CRC 的值由发送设备计算。接收设备在接收报文时重新计算 CRC 的值, 并将计算结果于实际接收到的 CRC 值相比较。如果两个值不相等, 则为错误。

CRC 的计算, 开始对一个 16 位寄存器预装全 1。然后将报文中的连续的 8 位子节对其进行后续的计算。只有字符中的 8 个数据位参与生成 CRC 的运算, 起始位, 停止位和校验位不参与 CRC 计算。

CRC 的生成过程中, 每个 8 - 位字符与寄存器中的值异或。然后结果向最低有效位 (LSB) 方向移动(Shift) 1 位, 而最高有效位 (MSB) 位置充零。然后提取并检查 LSB: 如果 LSB 为 1, 则寄存器中的值与一个固定的预置值异或; 如果 LSB 为 0, 则不进行异或操作。

这个过程将重复直到执行完 8 次移位。完成最后一次 (第 8 次) 移位及相关操作后, 下一个 8 位字节与寄存器的当前值异或, 然后又同上面描述过的一样重复 8 次。当所有报文中子节都运算之后得到的寄存器中的最终值, 就是 CRC。

生成 CRC 的过程为:

1. 将一个 16 位寄存器装入十六进制 FFFF (全 1)。将之称作 CRC 寄存器。
2. 将报文的第一个 8 位字节与 16 位 CRC 寄存器的低字节异或, 结果置于 CRC 寄存器。
3. 将 CRC 寄存器右移 1 位 (向 LSB 方向), MSB 充零。提取并检测 LSB。
4. (如果 LSB 为 0): 重复步骤 3 (另一次移位).
(如果 LSB 为 1): 对 CRC 寄存器异或多项式值 0xA001 (1010 0000 0000 0001)。
5. 重复步骤 3 和 4, 直到完成 8 次移位。当做完此操作后, 将完成对 8 位字节的完整操作。
6. 对报文中的下一个字节重复步骤 2 到 5, 继续此操作直至所有报文被处理完毕。
7. CRC 寄存器中的最终内容为 CRC 值。
8. 当放置 CRC 值于报文时, 如下面描述的那样, 高低字节必须交换。

将 CRC 放置于报文

当 16 位 CRC (2 个 8 位字节) 在报文中传送时, 低位字节首先发送, 然后是高位字节。

例如, 如果 CRC 值为十六进制 1241 (0001 0010 0100 0001):

