产品手册　　适用于工业专用设备

Modbus TCP 协议
编码手册

- **适用对象：**
  1. 程序开发人员；

- **适用场合：**
  1. 用户自己开发 PC 端 Modbus 通讯软件；
  2. 可以结合：Modscan32 软件说明书，开发后台软件更方便；

- **版本声明：**
  1. 该手册基于标准 Modbus 通讯协议，结合我司产品编写的开发手册；
  2. 目的是为了方便具有编码能力的客户设计开发自己的上位机软件；
  3. 如果您发现本手册中有错误或疑问，请与我们联系，谢谢；

- **电子手册：**

  在给您提供产品的同时，我们会提供包含产品的资料、工具软件等内容。

- **技术支持：**

  有关产品使用培训、技术咨询以及常见疑难问题，请与公司联系或到网站查询。

目录

1. Modbus TCP 报文格式

Modbus TCP 协议和 Modbus RTU 协议基本一样；

需熟悉以太网 TCP 的 Socket 链接，Socket 不属于 Modbus 介绍范畴；
本文只介绍 Modbus TCP 的数据包格式，和命令应答格式；

Modbus TCP 和 Modbus RTU 协议区别:
Modbus TCP 报文 = 6 字节协议头 + Modbus RTU 整个报文 - Modbus RTU 的两字节 CRC；
也就是说，Modbus TCP 比 Modbus RTU 多了 6 字节的头，少两字节的 CRC；

Modbus TCP 6 字节协议头格式如下：
txbuf[0] = 0 《事务标识符，固定 0 即可》
txbuf[1] = 0 《事务标识符，固定 0 即可》
txbuf[2] = 0 《协议标识符，固定 0 即可》
txbuf[3] = 0 《协议标识符，固定 0 即可》
txbuf[4] = 0 《长度高字节，固定 0 即可》
txbuf[5] = 《后面报文字节数》

txbuf[6]…….. txbuf[6 + txnum - 1] // 正规协议包，同 Modbus RTU 整个报文一样，无 CRC；

2. 读 AI 模拟量输入

举例：读取 8 路 AI 数据，模块地址=1：

Modbus TCP 格式： 《十六进制》

主机发送：**00 00 00 00 00 06** 01 04 00 00 00 08

从机应答：

00 00 00 00 00 13 01 04 10 A0 B0 A1 B1 A2 B2 A3 B3 A4 B4 A5 B5 A6 B6 A7 B7

报文详解：

主机发送的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	04H	读取寄存器 〈READ_INPUT〉
起始寄存器地址	2	0000H	寄存器地址： 0000H - 对应模拟量输入通道 0 **〈30001 寄存器〉** 。。。 。。。 0007H - 对应模拟量输入通道 7 **〈30008 寄存器〉** 该寄存器地址位于【3】区 数据发送顺序：高字节在前，如 0007，则顺序：00 07
读取寄存器数量	2	0008H	读取 8 个寄存器里的内容 **〈30001-30008 寄存器〉** 数据发送顺序：高字节在前，如 0008，则顺序：00 08

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	04H	读取寄存器 〈READ_INPUT〉
返回数据长度	1	10H	其中：10H = AI 个数 * 2 = 8 * 2
返回数据	16	21E6H 3B95H … … 0000H	0 通道，8678 1 通道，15253 … … 7 通道，0000 数据发送顺序：高字节在前，如 21E6，则顺序：21 E6

3. 读 DI 开关量输入

举例：读取 8 路 DI，模块地址=1:
Modbus TCP 格式： 《十六进制》
主机发送：**00 00 00 00 00 06 01 02 00 00 00 08**
从机应答：**00 00 00 00 00 04 01 02 01 02**

举例：读取 16 路 DI，模块地址=1:
Modbus TCP 格式： 《十六进制》
主机发送：**00 00 00 00 00 06 01 02 00 00 00 10**
从机应答：**00 00 00 00 00 04 01 02 02 03 FF**

举例：读取 18 路 DI，模块地址=1:
Modbus TCP 格式： 《十六进制》
主机发送：**00 00 00 00 00 06 01 02 00 00 00 12**
从机应答：**00 00 00 00 00 04 01 02 03 03 FF FF**

报文详解：

主机发送的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1	06	后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	02H	读取寄存器 <READ_STATE>
起始寄存器地址	2	0000H	0000H - 该寄存器对应 DI0 的开关状态 **<10001 寄存器>** 0001H - 该寄存器对应 DI1 的开关状态 **<10002 寄存器>** 。。。 。。。 0007H - 该寄存器对应 DI7 的开关状态 **<10008 寄存器>** 该寄存器地址位于【1】区 数据发送顺序：高字节在前，如 0007，则顺序：00 07
读取寄存器数量	2	0008H	读取 8 个开关量输入状态 **<10001-10008 寄存器>** 数据发送顺序：高字节在前，如 0008，则顺序：00 08

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1	04	后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	02H	读取寄存器 <READ_STATE>
返回字节长度	1	01H	返回 01 个字节的开关量输入状态 其中：01H = 开关量个数 / 8 = 8 / 8 如果个数>8，且<=16，则 01H 应该改为：02H 以此类推个数

返回数据	1	02H	02H 从低位到高位代表 DIO-DI7 的输入状态 02H 即表示：DI1 高电平 ON，其他低电平 OFF 如果个数>8，且<=16， 则字节数=2，发送报文=0102H， 数据发送顺序为高字节在前：01 02 01 即表示：DIO-DI7 02 即表示：DI8-DI15 以此类推个数

4. 写 AO 多路模拟量输出

举例：模块地址=1，设置：
AO0=10000，AO1=20000，AO2=30000， AO3=40000 〈十进制值〉
AO0= 2710，AO1= 4E20，AO2= 7530， AO3= 9C40 〈十六进制值〉

Modbus TCP 格式： 《十六进制》
主机发送：00 00 00 00 00 0F 01 10 00 00 00 04 08 27 10 4E 20 75 30 9C 40
从机应答：00 00 00 00 00 06 01 10 00 00 00 04

报文详解：

主机发送的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	10H	写多保持寄存器 〈WRITE_N_HOLD〉
起始寄存器地址	2	0000H	0000H -通道 0 **〈40001 寄存器〉** 。。。 。。。 0003H -通道 3 **〈40004 寄存器〉** 该寄存器地址位于【4】区 数据发送顺序：高字节在前，如 0003，则顺序：00 03
寄存器数量	2	0004H	4 个寄存器 AO0-3 **〈40001-40004 寄存器〉** **最多一次写 120 个寄存器** 数据发送顺序：高字节在前，如 0004，则顺序：00 04
数据字节长度	1	08H	寄存器数量 * 2
保持数据 AO0	2	2710H	写入 AO0 数据发送顺序：高字节在前，如 2710，则顺序：27 10
……	……	……	……
保持数据 AO3	2	9C40H	写入 AO3 数据发送顺序：高字节在前，如 9C40，则顺序：9C 40

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	10H	写多保持寄存器 〈WRITE_N_HOLD〉
起始寄存器地址	2	0000H	同“主机发送的报文格式”解析
寄存器数量	2	0004H	同“主机发送的报文格式”解析

5. 写 AO 单路模拟量输出

举例：模块地址=1：

Modbus TCP 格式： 《十六进制》

主机发送《AO-0 输出十进制 10000》：00 00 00 00 00 06 01 06 00 00 27 10

从机应答《AO-0 输出十进制 10000》：00 00 00 00 00 06 01 06 00 00 27 10

==========

主机发送《AO-1 输出十进制 10000》：00 00 00 00 00 06 01 06 00 01 27 10

从机应答《AO-1 输出十进制 10000》：00 00 00 00 00 06 01 06 00 01 27 10

==========

主机发送《AO-2 输出十进制 10000》：00 00 00 00 00 06 01 06 00 02 27 10

从机应答《AO-2 输出十进制 10000》：00 00 00 00 00 06 01 06 00 02 27 10

==========

主机发送《AO-3 输出十进制 10000》：00 00 00 00 00 06 01 06 00 03 27 10

从机应答《AO-3 输出十进制 10000》：00 00 00 00 00 06 01 06 00 03 27 10

==========

报文详解：

主机发送的报文格式：《设置模拟量输出 AO-0 通道》

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	06H	写寄存器 〈WRITE_1_HOLD〉
寄存器地址	2	0000H	0000H - 该寄存器对应通道 0 **〈40001 寄存器〉** 。。。 。。。 0003H - 该寄存器对应通道 3 **〈40004 寄存器〉** 该寄存器地址位于【4】区 数据发送顺序：高字节在前，如 0003，则顺序：00 03
写入数据	2	2710H	设置通道 0 的输出值为 10000

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	06H	写寄存器 〈WRITE_1_HOLD〉
寄存器地址	2	0000H	同“主机发送的报文格式”解析
写入数据	2	2710H	同“主机发送的报文格式”解析

6. 读 AO 模拟量输出

举例：读取 4 路 AO 数据，模块地址=1：

Modbus TCP 格式： 《十六进制》

主机发送：**00 00 00 00 00 06** 01 03 00 00 00 04

从机应答：

00 00 00 00 00 0D 01 03 08 21 E6 3B 95 11 22 00 00 7C 6D

报文详解：

主机发送的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	03H	读取寄存器 〈READ_HOLD〉
起始寄存器地址	2	0000H	0000H - 该寄存器对应通道 AO0 **〈40001 寄存器〉** 。。。 。。。 0003H - 该寄存器对应通道 AO3 **〈40004 寄存器〉** 该寄存器地址位于【4】区 数据发送顺序：高字节在前，如 0003，则顺序：00 03
读取寄存器数量	2	0004H	读取 4 个寄存器里的内容 **〈40001-40004 寄存器〉** 数据发送顺序：高字节在前，如 0004，则顺序：00 04

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	03H	读取寄存器 〈READ_HOLD〉
返回数据长度	1	08H	其中：08H = AO 个数 * 2 = 4 * 2
返回数据	8	21E6H 3B95H 0000H	0 通道，8678 1 通道，15253 ... ... 3 通道，0000 数据发送顺序：高字节在前，如 21E6，则顺序：21 E6

7. 写 DO 多路开关量输出

举例：模块地址=1:

Modbus TCP 格式： 《十六进制》

主机发送《DO-0、1 闭合，DO2-7 断开》：00 00 00 00 00 08 01 0F 00 00 00 08 01 03

从机应答《DO-0、1 闭合，DO2-7 断开》：00 00 00 00 00 06 01 0F 00 00 00 08

==========

主机发送《DO:0- 7 闭合》：00 00 00 00 00 08 01 0F 00 00 00 08 01 FF

从机应答《DO:0- 7 闭合》：00 00 00 00 00 06 01 0F 00 00 00 08 《TCP》

==========

主机发送《DO:0- 7 断开》：00 00 00 00 00 08 01 0F 00 00 00 08 01 00

从机应答《DO:0- 7 断开》：00 00 00 00 00 06 01 0F 00 00 00 08 《TCP》

==========

主机发送《DO:0-12 闭合》：00 00 00 00 00 09 01 0F 00 00 00 0C 02 FF FF

从机应答《DO:0-12 闭合》：00 00 00 00 00 06 01 0F 00 00 00 0C 《TCP》

==========

主机发送《DO:0-12 断开》：00 00 00 00 00 09 01 0F 00 00 00 0C 02 00 00

从机应答《DO:0-12 断开》：00 00 00 00 00 06 01 0F 00 00 00 0C 《TCP》

==========

主机发送《DO:0-15 闭合》：00 00 00 00 00 09 01 0F 00 00 00 10 02 FF FF

从机应答《DO:0-15 闭合》：00 00 00 00 00 06 01 0F 00 00 00 10 《TCP》

==========

主机发送《DO:0-15 断开》：00 00 00 00 00 09 01 0F 00 00 00 10 02 00 00

从机应答《DO:0-15 断开》：00 00 00 00 00 06 01 0F 00 00 00 10 《TCP》

==========

主机发送《DO:0-23 闭合》：00 00 00 00 00 0A 01 0F 00 00 00 18 03 FF FF FF

从机应答《DO:0-23 闭合》：00 00 00 00 00 06 01 0F 00 00 00 18 《TCP》

==========

主机发送《DO:0-23 断开》：00 00 00 00 00 0A 01 0F 00 00 00 18 03 00 00 00

从机应答《DO:0-23 断开》：00 00 00 00 00 06 01 0F 00 00 00 18 《TCP》

==========

主机发送《DO:0-31 闭合》：00 00 00 00 00 0B 01 0F 00 00 00 20 04 FF FF FF FF

从机应答《DO:0-31 闭合》：00 00 00 00 00 06 01 0F 00 00 00 20 《TCP》

==========

主机发送《DO:0-31 断开》：00 00 00 00 00 0B 01 0F 00 00 00 20 04 00 00 00 00

从机应答《DO:0-31 断开》：00 00 00 00 00 06 01 0F 00 00 00 20 《TCP》

==========

报文详解：

主机发送的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	0FH	写寄存器 <WRITE_N_COIL>
起始寄存器地址	2	0000H	起始寄存器地址： 0000H - 开关量输出通道 0 的开关状态**<00001 寄存器>** 。。。 。。。 0007H - 开关量输出通道 7 的开关状态**<00008 寄存器>** 该寄存器地址位于【0】区 **<00001-00008 寄存器>** 数据发送顺序：高字节在前，如 0007，则顺序：00 07
开关量输出 通道个数	2	0008H	8 个输出开关量输出
数据字节个数	1	01H	写入 01 个字节 其中：01H = 通道个数 / 8 = 8 / 8 如果个数>8，且<=16，则 01H 应该改为：02H 以此类推个数
写入数据	1	03H	03H 二进制按位表示：0 0 0 0 0 0 1 1 03H 从低位到高位代表 D00-D07 的开关状态 **03H 表示：D00、D01 闭合 ON，D02-7 断开 OFF** 如果个数>8，且<=16， 则字节数=2，**发送报文**=0302H， 数据发送顺序为高字节在前：03 02 03 即表示：D00-D07 02 即表示：D08-D015 以此类推个数

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	0FH	写寄存器 <WRITE_N_COIL>
起始寄存器地址	2	0000H	同“主机发送的报文格式”解析
开关量输出 通道个数	2	0008H	同“主机发送的报文格式”解析

8. 写 DO 单路开关量输出

举例：模块地址=1:

Modbus TCP 格式： 《十六进制》

主机发送《DO-0 闭合》：00 00 00 00 00 06 01 05 00 00 FF 00 《Modbus TCP》
从机应答《DO-0 闭合》：00 00 00 00 00 06 01 05 00 00 FF 00 《Modbus TCP》
主机发送《DO-0 断开》：00 00 00 00 00 06 01 05 00 00 00 00 《Modbus TCP》
从机应答《DO-0 断开》：00 00 00 00 00 06 01 05 00 00 00 00 《Modbus TCP》

==========

主机发送《DO-1 闭合》：00 00 00 00 00 06 01 05 00 01 FF 00 《Modbus TCP》
从机应答《DO-1 闭合》：00 00 00 00 00 06 01 05 00 01 FF 00 《Modbus TCP》
主机发送《DO-1 断开》：00 00 00 00 00 06 01 05 00 01 00 00 《Modbus TCP》
从机应答《DO-1 断开》：00 00 00 00 00 06 01 05 00 01 00 00 《Modbus TCP》

==========

主机发送《DO-2 闭合》：00 00 00 00 00 06 01 05 00 02 FF 00 《Modbus TCP》
从机应答《DO-2 闭合》：00 00 00 00 00 06 01 05 00 02 FF 00 《Modbus TCP》
主机发送《DO-2 断开》：00 00 00 00 00 06 01 05 00 02 00 00 《Modbus TCP》
从机应答《DO-2 断开》：00 00 00 00 00 06 01 05 00 02 00 00 《Modbus TCP》

==========

主机发送《DO-3 闭合》：00 00 00 00 00 06 01 05 00 03 FF 00 《Modbus TCP》
从机应答《DO-3 闭合》：00 00 00 00 00 06 01 05 00 03 FF 00 《Modbus TCP》
主机发送《DO-3 断开》：00 00 00 00 00 06 01 05 00 03 00 00 《Modbus TCP》
从机应答《DO-3 断开》：00 00 00 00 00 06 01 05 00 03 00 00 《Modbus TCP》

==========

主机发送《DO-4 闭合》：00 00 00 00 00 06 01 05 00 04 FF 00 《Modbus TCP》
从机应答《DO-4 闭合》：00 00 00 00 00 06 01 05 00 04 FF 00 《Modbus TCP》
主机发送《DO-4 断开》：00 00 00 00 00 06 01 05 00 04 00 00 《Modbus TCP》
从机应答《DO-4 断开》：00 00 00 00 00 06 01 05 00 04 00 00 《Modbus TCP》

==========

主机发送《DO-5 闭合》：00 00 00 00 00 06 01 05 00 05 FF 00 《Modbus TCP》
从机应答《DO-5 闭合》：00 00 00 00 00 06 01 05 00 05 FF 00 《Modbus TCP》
主机发送《DO-5 断开》：00 00 00 00 00 06 01 05 00 05 00 00 《Modbus TCP》
从机应答《DO-5 断开》：00 00 00 00 00 06 01 05 00 05 00 00 《Modbus TCP》

==========

主机发送《DO-6 闭合》：00 00 00 00 00 06 01 05 00 06 FF 00 《Modbus TCP》
从机应答《DO-6 闭合》：00 00 00 00 00 06 01 05 00 06 FF 00 《Modbus TCP》
主机发送《DO-6 断开》：00 00 00 00 00 06 01 05 00 06 00 00 《Modbus TCP》
从机应答《DO-6 断开》：00 00 00 00 00 06 01 05 00 06 00 00 《Modbus TCP》

==========

主机发送《DO-7 闭合》：00 00 00 00 00 06 01 05 00 07 FF 00 《Modbus TCP》
从机应答《DO-7 闭合》：00 00 00 00 00 06 01 05 00 07 FF 00 《Modbus TCP》
主机发送《DO-7 断开》：00 00 00 00 00 06 01 05 00 07 00 00 《Modbus TCP》
从机应答《DO-7 断开》：00 00 00 00 00 06 01 05 00 07 00 00 《Modbus TCP》

主机发送《DO-8 闭合》：00 00 00 00 00 06 01 05 00 08 FF 00 《Modbus TCP》
从机应答《DO-8 闭合》：00 00 00 00 00 06 01 05 00 08 FF 00 《Modbus TCP》
主机发送《DO-8 断开》：00 00 00 00 00 06 01 05 00 08 00 00 《Modbus TCP》
从机应答《DO-8 断开》：00 00 00 00 00 06 01 05 00 08 00 00 《Modbus TCP》

主机发送《DO-9 闭合》：00 00 00 00 00 06 01 05 00 09 FF 00 《Modbus TCP》
从机应答《DO-9 闭合》：00 00 00 00 00 06 01 05 00 09 FF 00 《Modbus TCP》
主机发送《DO-9 断开》：00 00 00 00 00 06 01 05 00 09 00 00 《Modbus TCP》
从机应答《DO-9 断开》：00 00 00 00 00 06 01 05 00 09 00 00 《Modbus TCP》

主机发送《DO-10 闭合》：00 00 00 00 00 06 01 05 00 0A FF 00 《Modbus TCP》
从机应答《DO-10 闭合》：00 00 00 00 00 06 01 05 00 0A FF 00 《Modbus TCP》
主机发送《DO-10 断开》：00 00 00 00 00 06 01 05 00 0A 00 00 《Modbus TCP》
从机应答《DO-10 断开》：00 00 00 00 00 06 01 05 00 0A 00 00 《Modbus TCP》

主机发送《DO-11 闭合》：00 00 00 00 00 06 01 05 00 0B FF 00 《Modbus TCP》
从机应答《DO-11 闭合》：00 00 00 00 00 06 01 05 00 0B FF 00 《Modbus TCP》
主机发送《DO-11 断开》：00 00 00 00 00 06 01 05 00 0B 00 00 《Modbus TCP》
从机应答《DO-11 断开》：00 00 00 00 00 06 01 05 00 0B 00 00 《Modbus TCP》

主机发送《DO-12 闭合》：00 00 00 00 00 06 01 05 00 0C FF 00 《Modbus TCP》
从机应答《DO-12 闭合》：00 00 00 00 00 06 01 05 00 0C FF 00 《Modbus TCP》
主机发送《DO-12 断开》：00 00 00 00 00 06 01 05 00 0C 00 00 《Modbus TCP》
从机应答《DO-12 断开》：00 00 00 00 00 06 01 05 00 0C 00 00 《Modbus TCP》

主机发送《DO-13 闭合》：00 00 00 00 00 06 01 05 00 0D FF 00 《Modbus TCP》
从机应答《DO-13 闭合》：00 00 00 00 00 06 01 05 00 0D FF 00 《Modbus TCP》
主机发送《DO-13 断开》：00 00 00 00 00 06 01 05 00 0D 00 00 《Modbus TCP》
从机应答《DO-13 断开》：00 00 00 00 00 06 01 05 00 0D 00 00 《Modbus TCP》

主机发送《DO-14 闭合》：00 00 00 00 00 06 01 05 00 0E FF 00 《Modbus TCP》
从机应答《DO-14 闭合》：00 00 00 00 00 06 01 05 00 0E FF 00 《Modbus TCP》
主机发送《DO-14 断开》：00 00 00 00 00 06 01 05 00 0E 00 00 《Modbus TCP》
从机应答《DO-14 断开》：00 00 00 00 00 06 01 05 00 0E 00 00 《Modbus TCP》

主机发送《DO-15 闭合》：00 00 00 00 00 06 01 05 00 0F FF 00 《Modbus TCP》
从机应答《DO-15 闭合》：00 00 00 00 00 06 01 05 00 0F FF 00 《Modbus TCP》
主机发送《DO-15 断开》：00 00 00 00 00 06 01 05 00 0F 00 00 《Modbus TCP》
从机应答《DO-15 断开》：00 00 00 00 00 06 01 05 00 0F 00 00 《Modbus TCP》

报文详解：

主机发送的报文格式：《设置 DO_0 闭合，模块地址=1:》

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	05H	写寄存器 <WRITE_1_COIL>
起始寄存器地址	2	0000H	寄存器地址： 0000H -开关量输出通道 0 的开关状态**<00001 寄存器>** 。。。 。。。 0007H -开关量输出通道 7 的开关状态**<00008 寄存器>** 该寄存器地址位于【0】区 数据发送顺序：高字节在前，如 0007，则顺序：00 07
写入数据	2	FF00H	将 FF00H 写入 0000H 寄存器中 FF00H：表示 DO 闭合 0000H：表示 DO 断开 数据发送顺序：高字节在前，如 FF00，则顺序：FF 00

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	05H	写寄存器 <WRITE_1_COIL>
起始寄存器地址	2	0000H	同“主机发送的报文格式”解析
写入数据	2	FF00H	同“主机发送的报文格式”解析

9. 读 DO 开关量输出

举例：读取 8 路 DO 状态，模块地址=1：

Modbus TCP 格式： 《十六进制》

主机发送：**00 00 00 00 00 06** 01 01 00 00 00 08

从机应答：**00 00 00 00 00 04** 01 01 01 03

举例：读取 12 路 DO 状态，模块地址=7：

Modbus TCP 格式： 《十六进制》

主机发送：**00 00 00 00 00 06** 07 01 00 00 00 0C

从机应答：**00 00 00 00 00 04** 07 01 02 03 FF

报文详解：

主机发送的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	01H	读取寄存器 <READ_COIL>
起始寄存器地址	2	0000H	0000H －开关量输出通道 0 状态 **<00001 寄存器>** 。。。 。。。 0007H － 开关量输出通道 7 状态 **<00008 寄存器>** 该寄存器地址位于【0】区 数据发送顺序：高字节在前，如 0007，则顺序：00 07
读取寄存器数量	2	0008H	读取 8 个寄存器里的内容 **<00001-00008 寄存器>** 数据发送顺序：高字节在前，如 0008，则顺序：00 08

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	01H	读取开关量输出 <READ_COIL>
返回数据长度	1	01H	其中：01H = 通道个数 / 8 = 8 / 8 如果个数>8，且<=16，则 01H 应该改为：02H 以此类推个数
返回数据	1	03H	03H 从低位到高位代表 DO0-DO7 的开关状态 03H 即表示：DO0、DO1 闭合 ON，DO2-7 断开 OFF 如果个数>8，且<=16， 则字节数=2，发送报文=0302H， 数据发送顺序为高字节在前：03 02

			03 即表示：D00-D07 02 即表示：D08-D015 以此类推个数

10.读 PT100 温度数值

举例：读取 8 路 PT100 数据，模块地址=1：

Modbus TCP 格式： 《十六进制》

主机发送：**00 00 00 00 00 06** 01 04 00 00 00 08

从机应答：

00 00 00 00 00 13 01 04 10 A0 B0 A1 B1 A2 B2 A3 B3 A4 B4 A5 B5 A6 B6 A7 B7

报文详解：

主机发送的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	04H	读取寄存器 <READ_INPUT>
起始寄存器地址	2	0000H	0000H - 该寄存器对应输入通道 0 **<30001 寄存器>** 。。。 。。。 0007H - 该寄存器对应输入通道 7 **<30008 寄存器>** 该寄存器地址位于【3】区 数据发送顺序：高字节在前，如 0007，则顺序：00 07
读取寄存器数量	2	0008H	读取 8 个寄存器里的内容 **<30001-30008 寄存器>** 数据发送顺序：高字节在前，如 0008，则顺序：00 08

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	04H	读取寄存器 <READ_INPUT>
返回数据长度	1	10H	其中：10H = 通道个数 * 2 = 8 * 2
返回数据	16	21E6H 3B95H 0000H	0 通道，8678 1 通道，15253 ... ... 7 通道，0000 数据发送顺序：高字节在前，如 21E6，则顺序：21 E6

11.读 PI 脉冲输入

举例：读取 1 路 PI 数据，寄存器地址 30041,30042，模块地址=1：

Modbus TCP 格式： 《十六进制》

主机发送：**00 00 00 00 00 06** 01 04 00 28 00 02

从机应答：

00 00 00 00 00 13 01 04 04 A0 B0 A1 B1 61 87

报文详解：

主机发送的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	04H	读取寄存器 <READ_INPUT>
起始寄存器地址	2	0028H	寄存器地址： 0028H - 对应 **<30041 寄存器>** 该寄存器地址位于【3】区 数据发送顺序：高字节在前，如 0028，则顺序：00 28
读取寄存器数量	2	0002H	读取 2 个寄存器里的内容 **<30041-30042 寄存器>** 数据发送顺序：高字节在前，如 0002，则顺序：00 02

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	04H	读取寄存器 <READ_INPUT>
返回数据长度	1	04H	其中：04H = 寄存器个数 * 2 = 2 * 2
返回数据	4	21E6H 3B95H	8678 **<30041 寄存器>** 15253 **<30042 寄存器>** 数据发送顺序：高字节在前，如 21E6，则顺序：21 E6

12.PI 脉冲个数清零

举例：Modbus 寄存器地址=00048，模块地址=1:

Modbus TCP 格式： 《十六进制》

主机发送：**00 00 00 00 00 06** 01 05 00 2F FF 00 《Modbus TCP》

从机应答：**00 00 00 00 00 06** 01 05 00 2F FF 00 《Modbus TCP》

报文详解：

主机发送的报文格式： 《设置 DO_0 闭合，模块地址=1:》

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	05H	写寄存器 <WRITE_1_COIL>
起始寄存器地址	2	002FH	寄存器地址： 002FH - **<00048 寄存器>** 该寄存器地址位于【0】区 数据发送顺序：高字节在前，如 002F，则顺序：00 2F
写入数据	2	FF00H	将 FF00H 写入 0000H 寄存器中,清除脉冲累计值 数据发送顺序：高字节在前，如 FF00，则顺序：FF 00

从机应答的报文格式：

发送内容	字节数	发送报文	备注
协议头部	5	0 0 0 0 0	固定全为 0 即可
下面字节数	1		后面报文字节数
模块地址	1	01H	模块地址 = 1
功能码	1	05H	写寄存器 <WRITE_1_COIL>
起始寄存器地址	2	002FH	同“主机发送的报文格式”解析
写入数据	2	FF00H	同“主机发送的报文格式”解析