综科智控

MODBUS-RTU/TCP

协议详解

【】绝密

【X】NDA

【】公开

深圳市综科智控科技开发有限公司

——综合科技，智慧控制

版本历史			
版本	修订日期	修订人	修订内容
1.0	2013-10-25	综科智控	初稿发布
1.1	2014-1-05	综科智控	修正文字错误
1.2	2019-11-05	综科智控	将复杂的协议介绍简单化

目录

一、MODBUS 协议简介

1. 什么是 MODBUS 协议

Modbus 协议是一种已广泛应用于当今工业控制领域的通用通讯协议，按其格式可分为 MODBUS-RTU,MODBUS-ASCII,MODBUS-TCP,其中前两者适用于串行通信控制网络中,例如 RS485,RS232 等,而 MODBUS-TCP 主要应用于基于以太网 TCP/IP 通信的控制网络中。

通过此协议，控制器相互之间、或控制器经由网络(如以太网)可以和其它设备之间进行通信。Modbus 协议使用的是主从通讯技术，即由主设备主动查询和操作从设备。一般将主控设备方所使用的协议称为 Modbus Master，从设备方使用的协议称为 Modbus Slave。典型的主设备包括工控机和工业控制器等;典型的从设备如 PLC 可编程控制器等。Modbus 通讯物理接口可以选用串口(包括 RS232 和 RS485)，也可以选择以太网口。其通信遵循以下的过程：

- 主设备向从设备发送请求
- 从设备分析并处理主设备的请求，然后向主设备发送结果
- 如果出现任何差错，从设备将返回一个异常功能码

Modbus 协议具有以下几个特点：

(1)标准、开放，用户可以免费、放心地使用 Modbus 协议，不需要交纳许可证费，也不会侵犯知识产权。

(2)Modbus 可以支持多种电气接口，如 RS-232、RS-485 等，还可以在各种介质上传送，如双绞线、光纤、无线等。

(3)Modbus 的帧格式简单、紧凑，通俗易懂。用户使用容易，厂商开发简单。

2. MODBUS 协议格式

2.1 MODBUS-RTU 报文模型(用于串口通讯)

设备地址	功能码	数据	CRC 校验 L	CRC 校验 H
1BYTE	1BYTE	N*BYTE	1BYTE	1BYTE

2.2 MODBUS-TCP 报文模型(用于网络 TCP/IP 通信)

传输标识符 TID	协议标识符 PID	后面要传输的字节数	设备地址	功能码	数据
2BYTE	2BYTE	2BYTE	1BYTE	1BYTE	N*BYTE

注意：
Modbus-RTU 一般用于串口 RS232/RS485/RS422 通讯
Modbus-TCP 一般用于网口、WIFI 的 TCP/UDP 通讯

3. MODBUS 的功能码及寄存器介绍

3.1 功能码

下表列出 MODBUS 支持的部分功能代码：以十进制表示

功能码(十进制)	作用	信息地址	位操作/字操作	操作数量
01	读单个/多个 DO 寄存器	00001-09999	位操作	单个/多个
02	读单个/多个 DI 寄存器	10001-19999	位操作	单个/多个
03	读单个/多个 AO 寄存器	40001-49999	字操作	单个/多个
04	读单个/多个 AI 寄存器	30001-39999	字操作	单个/多个
05	写单个 DO 寄存器	00001-09999	位操作	单个
06	写单个 AO 寄存器	40001-49999	字操作	单个
15	写单个/多个 DO 寄存器	00001-09999	位操作	单个/多个
16	写单个/多个 AO 寄存器	40001-49999	字操作	单个/多个

3.2 寄存器分类说明

寄存器种类	说明
DI 寄存器	只读，主要用于读模块的 DI 离散量输入信号的 ON/OFF 状态。
DO 寄存器	可读/可写，主要用于控制相应 DO 输出，例如：Y 点的开关状态。也可以用于控制某些功能的启动或停止、打开或关闭，例如：PWM 的输出/停止，计数的清空等。
AI 寄存器	只度，主要用于读取模块中的输入型数据,例如 AD 模拟采集到的电压值,电流值,压力值等,或者 X 输入脉冲计数值等。
AO 寄存器	可读/可写，主要用于存放的用户下发的参数,例如 AO 模拟量输出值、PWM 输出频率、占空比等。

3.3 寄存器地址说明

■ 寄存器信息地址（PLC 地址）：

寄存器信息地址指的是存放于控制器中的地址，这些控制器可以是 PLC，也可以使触摸屏，或是文本显示器。例如 4x0001、3x0002 等，这些地址一般使用十进制描述。

■ 寄存器寻址地址（协议地址）：

寄存器寻址地址指的是通信时使用的寄存器地址，例如信息地址 40001 对应寻址地址 0x0000，40002 对应寻址地址 0x0001，寄存器寻址地址一般使用 16 进制描述。再如，信息寄存器 40003 对应寻址地址 0002，信息寄存器 30003 对应寻址地址 0002，虽然两个信息寄存器通信时使用相同的地址，但是需要使用不同的命令才可以访问，所以访问时不存在冲突。

二、MODBUS-RTU 协议详解

1. X 输入口开关量状态读取 (读取:DI 寄存器,命令号:0x02)

■ 描述

读模块的 X 输入口输入信号的 ON/OFF 状态。

■ 例子：读 X1 当前输入状态的请求&响应报文

请求：

发送数据(HEX): 01 02 00 00 00 01 B9 CA

解释：

01:子站地址

02:指令号，02 读 DI

00 00:从哪一路开始读，0x0000=X1 开始读

00 01:要读多少路，十六进制 0x0001=十进制 1，读取 1 路

B9 CA:CRC 校验

响应：

//X1 输入 ON(1)时的回复

01 02 01 01 60 48

解释：

01:子站地址

02:指令号，02 读 DI

01:后面跟的数据字节数

01:换成二进制就是 0000 0001 =X8-X7-X6-X5-X4-X3-X2-X1 输入口的当前状态 0=OFF，1=ON

60 48:CRC 校验

//X1 输入 OFF(0)时的回复

01 02 01 00 A1 88

解释：

01:子站地址

02:指令号，02 读 DI

01:后面跟的数据字节数

00:换成二进制就是 0000 0000 =X8-X7-X6-X5-X4-X3-X2-X1 输入口的当前状态 0=OFF，1=ON

A1 88:CRC 校验

■ 例子：读 X1-X8 当前输入状态的请求&响应报文

请求：

发送数据(HEX): 01 02 00 00 00 08 79CC

解释：

01:子站地址

02:指令号，02 读 DI

00 00:从哪一路开始读，0x0000=X1 开始读

00 08:要读多少路十六进制 0x0008=十进制 8，读取 8 路

79CC:CRC 校验

响应：

//X1,X8 输入 ON(1)时,其他都输入 OFF(0)时的回复

01 02 01 81 61 E8

解释：

01:子站地址

02:指令号，02 读 DI

01:后面跟的数据字节数

81:换成二进制就是 1000 0001 =X8-X7-X6-X5-X4-X3-X2-X1 输入口的当前状态 0=OFF，1=ON

61 E8:CRC 校验

■ 例子：读 X1-X24 当前输入状态的请求&响应报文

请求：

发送数据(HEX): 01 02 00 00 00 18 78 00

解释：

01:子站地址

02:指令号，02 读 DI

00 00:从哪一路开始读，0x0000=X1 开始读

00 18:要读多少路十六进制 0x0018=十进制 24，读取 24 路

78 00:CRC 校验

响应：

//X1,X2,X9,X24 输入 ON(1)时,其他都输入 OFF(0)时的回复

01 02 03 03 01 80 88 7E

解释：

01:子站地址

02:指令号，02 读 DI

03:后面跟的数据字节数

03:换成二进制就是 0000 0011 =X8-X7-X6-X5-X4-X3-X2-X1 输入口的当前状态 0=OFF，1=ON

01:换成二进制就是 0000 0001 =X16-X15-X14-X13-X12-X11-X10-X9 输入口的当前状态 0=OFF，1=ON

80:换成二进制就是 1000 0000 =X24-X23-X22-X21-X20-X19-X18-X17 输入口的当前状态 0=OFF，1=ON

88 7E:CRC 校验

2. X 输入口脉冲计数读取 (读取:AI 寄存器,命令号:0x04)

■ 描述

读模块的 X 输入口的脉冲计数值。

■ 例子：读当前 X1 输入点脉冲计数值的请求&响应报文

请求：

发送数据(HEX): 01 04 00 18 00 02 F1 CC

解释：

01:子站地址

04:指令号，04 读 AI

00 18:从哪一路开始读，0x0018 对应的是 X1 脉冲计数寄存器起始地址

00 02:要读多少个寄存器，因为一路 X 脉冲计数占用 2 个寄存器地址，所以这里寄存器数量要填 2，十六进制 0x0002=十进制 2

F1 CC:CRC 校验

响应：

接收数据(HEX): 01 04 04 00 00 27 10 E1 B8

解释：

01:子站地址

04:指令号，04 读 AI

04:后面跟的数据字节数,0x04=十进制 4，后面数据区有 4 字节数据

00002710:换成十进制就是 0x00002710 =10000,即读取到 X1 输入口当前脉冲计数值为 10000

E1 B8:CRC 校验

■ 例子：读当前 X1-X4 输入点脉冲计数值的请求&响应报文

请求：

发送数据(HEX): 01 04 00 18 00 08 71 CB

解释：

01:子站地址

04:指令号，04 读 AI

00 18:从哪一路开始读，0x0018 对应的是 X1 脉冲计数寄存器起始地址

00 08:要读多少个寄存器，因为一路 X 脉冲计数占用 2 个寄存器地址，这里读 4 路，所以这里寄存器数量要填 8，十六进制 0x0008=十进制 8

71 CB :CRC 校验

响应：

接收数据(HEX): 01 04 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 EC C3

解释：

01:子站地址

04:指令号，04 读 AI

10:后面跟的数据字节数,0x10=十进制 16，后面数据区有 16 字节数据

00002710:换成十进制就是 0x00002710 =10000,即读取到 X1 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X2 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X3 输入口当前脉冲计数值为 10000
00002710:换成十进制就是 0x00002710 =10000,即读取到 X4 输入口当前脉冲计数值为 10000
EC C3:CRC 校验

■ 例子：读当前 X1-X8 输入点脉冲计数值的请求&响应报文

请求：

发送数据(HEX): 01 04 00 18 00 10 71 C1

解释：

01:子站地址

04:指令号，04 读 AI

00 18:从哪一路开始读，0x0018 对应的是 X1 脉冲计数寄存器起始地址

00 10:要读多少个寄存器，因为一路 X 脉冲计数占用 2 个寄存器地址，这里读 8 路，所以这里寄存器数量要填十六进制 0x0010=十进制 16

71 C1 :CRC 校验

响应：

接收数据(HEX): 01 04 20 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 36 38

解释：

01:子站地址

04:指令号，04 读 AI

20:后面跟的数据字节数,0x20=十进制 32，后面数据区有 32 字节数据

00002710:换成十进制就是 0x00002710 =10000,即读取到 X1 输入口当前脉冲计数值为 10000
00002710:换成十进制就是 0x00002710 =10000,即读取到 X2 输入口当前脉冲计数值为 10000
00002710:换成十进制就是 0x00002710 =10000,即读取到 X3 输入口当前脉冲计数值为 10000
00002710:换成十进制就是 0x00002710 =10000,即读取到 X4 输入口当前脉冲计数值为 10000
00002710:换成十进制就是 0x00002710 =10000,即读取到 X5 输入口当前脉冲计数值为 10000
00002710:换成十进制就是 0x00002710 =10000,即读取到 X6 输入口当前脉冲计数值为 10000
00002710:换成十进制就是 0x00002710 =10000,即读取到 X7 输入口当前脉冲计数值为 10000
00002710:换成十进制就是 0x00002710 =10000,即读取到 X8 输入口当前脉冲计数值为 10000
36 38:CRC 校验

■ 例子：读当前 X1-X12 输入点脉冲计数值的请求&响应报文

请求：

发送数据(HEX): 01 04 00 18 00 18 70 07

解释：

01:子站地址

04:指令号，04 读 AI

00 18:从哪一路开始读，0x0018 对应的是 X1 脉冲计数寄存器起始地址

00 18:要读多少个寄存器，因为一路 X 脉冲计数占用 2 个寄存器地址，这里读 12 路，所以这里寄存器数量要填十六进制 0x0018=十进制 24

70 07 :CRC 校验

响应：

接收数据(HEX): 01 04 30 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27

10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 C1 D9

解释：

01:子站地址

04:指令号，04 读 AI

30:后面跟的数据字节数,0x30=十进制 48，后面数据区有 48 字节数据

00002710:换成十进制就是 0x00002710 =10000,即读取到 X1 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X2 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X3 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X4 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X5 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X6 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X7 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X8 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X9 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读到 X10 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读到 X11 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读到 X12 输入口当前脉冲计数值为 10000

C1 D9:CRC 校验

3. X 输入口脉冲计数清空 (写入:DO 寄存器,命令号:0x0F)

■ 描述

清空模块的 X 输入口的脉冲计数值。

■ 例子：清空 X1 输入计数值的请求&响应报文

请求：

发送数据(HEX): 01 0f 00 40 00 01 01 01 ee 98

解释：

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 01:要清空多少路计数，十六进制 0x0001=十进制 1 路

01:后面要写入的数据字节数,0x01=十进制 1，写入 1 字节数据

01:写入的数据，写 1 清空

ee 98:CRC 校验

响应：

//模块回复

01 0F 00 40 00 01 95 df

解释：

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 01:要清空多少路计数，十六进制 0x0001=十进制 1 路

95 df:CRC 校验

■ 例子：清空 X1-X12 输入计数值的请求&响应报文

请求：

发送数据(HEX): 01 0f 00 40 00 0C 02 ff 0f ea 84

解释：

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 0C:要清空多少路计数，十六进制 0x000C=十进制 12 路

02:后面要写入的数据字节数,0x02=十进制 2，写入 2 字节数据

ff:十六进制 0xff=二进制 1111 1111=X8 计数清空-X7 计数清空-X6 计数清空-X5 计数清空-X4 计数清空-X3 计数清空-X2 计数清空-X1 计数清空，对应的 bit 位写 1 清空，写 0 保持不变

0f: 十六进制 0x0f=二进制 0000 1111=X16 计数清空-X15 计数清空-X14 计数清空-X13 计数清空-X12 计数清空-X11 计数清空-X10 计数清空-X9 计数清空，对应的 bit 位写 1 清空，写 0 保持不变

ea 84:CRC 校验

响应：

//模块回复

01 0F 00 40 00 0C 54 1A

解释：

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 0C:要清空多少路计数，十六进制 0x000C=十进制 12 路

54 1A:CRC 校验

■ 例子：指定清空 X1,X4,X8,X12,X24 输入计数值的请求&响应报文

请求：

发送数据(HEX): 01 0f 00 40 00 18 03 85 08 80 16 39

解释：

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 18:要清空多少路计数，十六进制 0x0018=十进制 24 路

03:后面要写入的数据字节数,0x03=十进制 3，写入 3 字节数据

85:十六进制 0x85=二进制 1000 1001=X8 计数清空-X7 计数清空-X6 计数清空-X5 计数清空-X4 计数清空-X3 计数清空-X2 计数清空-X1 计数清空，对应的 bit 位写 1 清空，写 0 保持不变

08: 十六进制 0x08=二进制 0000 1000=X16 计数清空-X15 计数清空-X14 计数清空-X13 计数清空-X12 计数清空-X11 计数清空-X10 计数清空-X9 计数清空，对应的 bit 位写 1 清空，写 0 保持不变

80: 十六进制 0x80=二进制 1000 0000=X24 计数清空-X23 计数清空-X22 计数清空-X21 计数清空-X20 计数清空-X19 计数清空-X18 计数清空-X17 计数清空，对应的 bit 位写 1 清空，写 0 保持不变

16 39:CRC 校验

响应：

//模块回复

01 0F 00 40 00 18 54 15

解释：

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 18:操作了多少路计数，十六进制 0x0018=十进制 24 路

54 15:CRC 校验

4. Y 输出口 ON/OFF 写入 (写入:DO 寄存器,命令号:0x0F)

■ 描述

用于控制模块 Y 输出点的开关状态。

■ 例子：控制 Y1 当前输出 ON/OFF 的请求&响应报文

请求：

发送数据(HEX): 01 0f 00 00 00 01 01 01 ef 57

解释：
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 01:要写多少路，十六进制 0x0001=十进制 1，写 1 路
01:后面要写入的数据字节数,0x01=十进制 1，写入 1 字节数据
01:写入的数据，0x01 换算成二进制 00000001 = Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 输出状态，ON=0，OFF=1
ef 57:CRC 校验

响应：

//模块回复
01 0F 00 00 00 01 94 0B

解释：
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 01:写了多少路，十六进制 0x0001=十进制 1，写 1 路
94 0B:CRC 校验

■ 例子：控制 Y1-Y8 当前输出 ON/OFF 的请求&响应报文

//将 Y1，Y2，Y8 输出 ON，其他路输出 OFF

请求：

发送数据(HEX): 01 0f 00 00 00 08 01 83 bf 34

解释：
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 08:要写多少路，十六进制 0x0008=十进制 8，写 8 路
01:后面要写入的数据字节数,0x01=十进制 1，写入 1 字节数据

83:写入的数据，0x83 换算成二进制 10000011 = Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 输出状态，ON=0，OFF=1
bf 34:CRC 校验

响应：
//模块回复
01 0F 00 00 00 08 54 0D
解释：
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 08:写了多少路，十六进制 0x0008=十进制 8，写 8 路
54 0D:CRC 校验

■ 例子：控制 Y1-Y24 当前输出 ON/OFF 的请求&响应报文

//将 Y1，Y2，Y9，Y24 输出 ON，其他路输出 OFF
请求：
发送数据(HEX): 01 0f 00 00 00 18 03 03 01 80 b0 44
解释：
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 18:要写多少路，十六进制 0x0018=十进制 24，写 24 路
03:后面要写入的数据字节数,0x01=十进制 1，写入 1 字节数据
03:写入的数据，0x03 换算成二进制 00000011 = Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 输出状态，ON=0，OFF=1
01:写入的数据，0x01 换算成二进制 00000001 = Y16-Y15-Y14-Y13-Y12-Y11-Y10-Y9 输出状态，ON=0，OFF=1
80:写入的数据，0x80 换算成二进制 10000000 = Y24-Y23-Y22-Y21-Y20-Y19-Y18-Y17 输出状态，ON=0，OFF=1
b0 44:CRC 校验

响应：
//模块回复
01 0F 00 00 00 18 55 C1
解释：
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 18:写了多少路，十六进制 0x0018=十进制 24，写 24 路
55 C1:CRC 校验

5. Y 输出口 ON/OFF 读取 (读取:DO 寄存器,命令号:0x01)

■ 描述

用于读取当前 Y 输出口的开关状态。

■ 例子：读 Y1 当前输出状态的请求&响应报文

请求：

发送数据(HEX): 01 01 00 00 00 01 FD CA

解释：

01:子站地址

01:指令号，01 读取多路 DO

00 00:从哪一路开始读，00 00=Y1 开始读

00 01:要读多少路，十六进制 0x0001=十进制 1，读取 1 路

FD CA:CRC 校验

响应：

//Y1 输出 ON(1)时的回复

01 01 01 01 90 48

解释：

01:子站地址

01:指令号，01 读取多路 DO

01:后面数据区字节数

01:数据，换算成二进制 0x01=0000 0001,对应 Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 的状态

90 48 :CRC 校验

//Y1 输出 OFF(0)的时回复

01 01 01 00 51 88

解释：

01:子站地址

01:指令号，01 读取多路 DO

01:后面数据区字节数

00:数据，换算成二进制 0x00=0000 0000,对应 Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 的状态

51 88 :CRC 校验

■ 例子：读 Y1-Y8 当前输出状态的请求&响应报文

请求：

发送数据(HEX): 01 01 00 00 00 08 3D CC

解释：

01:子站地址

01:指令号，01 读取多路 DO

00 00:从哪一路开始读，00 00=Y1 开始读

00 08:要读多少路，十六进制 0x0008=十进制 8，读取 8 路

3D CC:CRC 校验

响应：

//Y1,Y2,Y8 输出 ON(1)，其他输出 OFF(0)时的回复

01 01 01 83 10 29

解释：

01:子站地址

01:指令号，01 读取多路 DO

01:后面数据区字节数,0x01=十进制 1，后面数据区有 1 字节数据

83:数据区，换算成二进制 0x83=1000 0011,对应 Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 的状态

10 29 :CRC 校验

■ 例子：读 Y1-Y24 当前输出状态的请求&响应报文

请求：

发送数据(HEX): 01 01 00 00 00 18 3C 00

解释：

01:子站地址

01:指令号，01 读取多路 DO

00 00:从哪一路开始读，00 00=Y1 开始读

00 18:要读多少路，十六进制 0x0018=十进制 24，读取 24 路

3C00:CRC 校验

响应：

//Y1,Y2,Y9,Y24 输出 ON(1)，其他输出 OFF(0)时的回复

01 01 03 03 01 80 CC 7E

解释：

01:子站地址

01:指令号，01 读取多路 DO

03:后面数据区字节数,0x03=十进制 3，后面数据区有 3 字节数据

03:数据，换算成二进制 0x03=0000 0011,对应 Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 的状态

01:数据，换算成二进制 0x01=0000 0001,对应 Y16-Y15-Y14-Y13-Y12-Y11-Y10-Y9 的状态

80:数据，换算成二进制 0x80=1000 0000,对应 Y24-Y23-Y22-Y21-Y20-Y19-Y18-Y17 的状态

CC 7E :CRC 校验

6. AI 模拟量采集读取(读取:AI 寄存器,命令号:0x04)

■ 描述

读模块中的输入型数据,例如 AD 模拟采集到的电压值,电流值,压力值等。

■ 例子：读当前 AI 通道 AI1 模拟量输入值的请求&响应报文

请求：

发送数据(HEX): 01 04 00 00 00 02 71CB

解释：

01:子站地址

04:指令号，04 读 AI

00 00:从哪一路开始读，0x0000=AIO1 开始读

00 02:要多少个寄存器，因为一个 AI 通道占用 2 个寄存器地址，所以这里读 1 路模拟量的话，寄存器数量要填 2，十六进制 0x0002=十进制 2

71CB:CRC 校验

响应：

接收数据(HEX): 01 04 04 00 00 BF 11 4B B8

解释：

01:子站地址

04:指令号，04 读 AI

04:后面跟的数据字节数,0x04=十进制 4，后面数据区有 4 字节数据

0000BF11:换成十进制就是 0x0000BF11 =48913,即读取到 AI1 模拟量输入口当前值为 48913

4B B8:CRC 校验

■ 例子：读当前 AI 通道 AI1-AI4 模拟量输入值的请求&响应报文

请求：

发送数据(HEX): 01 04 00 00 00 08 F1 CC

解释：

01:子站地址

04:指令号，04 读 AI

00 00:从哪一路开始读，0x0000=AIO1 开始读

00 08:要多少个寄存器，因为一个 AI 通道占用 2 个寄存器地址，所以这里读 4 路模拟量的话，寄存器数量要填 8，十六进制 0x0008=十进制 8

F1 CC:CRC 校验

响应：

接收数据(HEX): 01 04 10 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 76 C5

解释：

01:子站地址

04:指令号，04 读 AI

10:后面跟的数据字节数,0x10=十进制 16，后面数据区有 16 字节数据

00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI1 模拟量输入口当前值

为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI2 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI3 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI4 模拟量输入口当前值为 48913
76 C5:CRC 校验

■ 例子：读当前 AI 通道 AI1-AI8 模拟量输入值的请求&响应报文

请求：

发送数据(HEX): 01 04 00 00 00 10 F1 C6
解释：
01:子站地址
04:指令号，04 读 AI
00 00:从哪一路开始读，0x0000=AIO1 开始读
00 10:要多少个寄存器，因为一个 AI 通道占用 2 个寄存器地址，所以这里读 8 路模拟量的话，寄存器数量要填 16，换算成十六进制就是 0x0010
F1 C6:CRC 校验

响应：

接收数据(HEX): 01 04 20 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 C0 F7
解释：
01:子站地址
04:指令号，04 读 AI
20:后面跟的数据字节数,0x20=十进制 32，后面数据区有 32 字节数据
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI1 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI2 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI3 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI4 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI5 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI6 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI7 模拟量输入口当前值为 48913
00 00 BF 11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI8 模拟量输入口当前值为 48913
C0 F7:CRC 校验

7. AO 模拟量输出写入(写入:AO 寄存器,命令号:0x10)

■ 描述

用于设定 DA 模拟量输出值。

■ 例子：写当前 AO 通道 AO1 模拟量输出值的请求&响应报文

请求：

发送数据(HEX): 01 10 00 00 00 01 02 10 D2 2B CD

解释：
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 01:要写多少路，十六进制 0x0001=十进制 1，写 1 路
02:后面要写入的数据字节数，0x02=十进制 2，写 2 个字节
10 D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO1 输出口模拟量输出值为 4306
2B CD:CRC 校验

响应：

接收数据(HEX): 01 10 00 00 00 01 01 C9

解释：
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 01:写了多少路，十六进制 0x0001=十进制 1，写 1 路
01 C9:CRC 校验

■ 例子：写当前 AO 通道 AO1-AO4 模拟量输出值的请求&响应报文

请求：

发送数据(HEX): 01 10 00 00 00 04 08 10 D2 10 D2 10 D2 10 D2 D6 9D

解释：
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 04:要写多少路，十六进制 0x0004=十进制 4，写 4 路
08:后面要写入的数据字节数，0x08=十进制 8，写 8 个字节
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO1 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO2 输出口模拟量输出值为 4306

10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO3 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO4 输出口模拟量输出值为 4306
D6 9D:CRC 校验

响应：
接收数据(HEX): 01 10 00 00 00 04 C1 CA
解释：
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 04:写了多少路，十六进制 0x0004=十进制 4，写 4 路
C1 CA:CRC 校验

■ 例子：写当前 AO 通道 AO1-AO8 模拟量输出值的请求&响应报文

请求：
发送数据(HEX): 01 10 00 00 00 08 10 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2 E8 0C
解释：
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 08:要写多少路，十六进制 0x0008=十进制 8，写 8 路
10:后面要写入的数据字节数，0x10=十进制 16，16 个字节
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO1 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO2 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO3 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO4 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO5 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO6 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO7 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO8 输出口模拟量输出值为 4306
E8 0C:CRC 校验

响应：
接收数据(HEX): 01 10 00 00 00 08 C1 CF

解释：
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 08:写了多少路，十六进制 0x0008=十进制 8，写 8 路
C1 CF:CRC 校验

8. AO 模拟量输出读取(读取:AO 寄存器,命令号:0x03)

■ 描述

用于读取当前 DA 模拟量输出值。

■ 例子：读当前 AO 通道 AO1 模拟量输出值的请求&响应报文

请求：

发送数据(HEX): 01 03 00 00 00 01 84 0A

解释：
01:子站地址
03:指令号，03 读 AO
00 00:从哪一路开始读，0x0000=AIO1 开始读
00 01:要读多少路，十六进制 0x0001=十进制 1，读取 1 路
84 0A:CRC 校验

响应：

接收数据(HEX): 01 03 02 10 D2 35 D9

解释：
01:子站地址
03:指令号，03 读 AO
02:后面跟的数据字节数
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO1 输出模拟量当前输出值为 4306
35 D9:CRC 校验

■ 例子：读当前 AO 通道 AO1-AO4 模拟量输出值的请求&响应报文

请求：

发送数据(HEX): 01 03 00 00 00 04 44 09

解释：
01:子站地址
03:指令号，03 读 AO
00 00:从哪一路开始读，0x0000=AIO1 开始读
00 04:要读多少路，十六进制 0x0004=十进制 4，读取 4 路
44 09:CRC 校验

响应：

接收数据(HEX): 01 03 08 10 D2 10 D2 10 D2 10 D2 F5 30

解释：
01:子站地址
03:指令号，03 读 AO
08:后面跟的数据字节数
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO1 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO2 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO3 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO4 输出模拟量当前输出值为 4306

F5 30:CRC 校验

■ 例子：读当前 AO 通道 AO1-AO8 模拟量输出值的请求&响应报文

请求：
发送数据(HEX): 01 03 00 00 00 08 44 0C

解释：
01:子站地址
03:指令号，03 读 AO
00 00:从哪一路开始读，0x0000=AIO1 开始读
00 08:要读多少路，十六进制 0x0008=十进制 8，读取 8 路
44 0C:CRC 校验

响应：
接收数据(HEX): 01 03 10 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2 15 98

解释：
01:子站地址
03:指令号，03 读 AO
10:后面跟的数据字节数,十六进制 0x10=十进制 16
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO1 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO2 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO3 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO4 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO5 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO6 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO7 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO8 输出模拟量当前输出值为 4306
15 98:CRC 校验

三、MODBUS-TCP 协议详解

1. X 输入口开关量状态读取 (读取:DI 寄存器,命令号:0x02)

■ 描述

读模块 X 输入点的 ON/OFF 状态等。

■ 例子：读 X1 当前输入状态的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 02 00 00 00 01

解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
02:指令号，02 读 DI
00 00:从哪一路开始读，0x0000=X1 开始读
00 01:要读多少路，十六进制 0x0001=十进制 1，读取 1 路

响应：

//X1 输入 ON(1)时的回复
00 00 00 00 00 04 01 02 01 01

解释：
00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 04：后面要发送的字节数
01:子站地址
02:指令号，02 读 DI
01:后面跟的数据字节数
01:换成二进制就是 0000 0001 =X8-X7-X6-X5-X4-X3-X2-X1 输入口的当前状态 0=OFF，1=ON

//X1 输入 OFF(0)时的回复
00 00 00 00 00 04 01 02 01 00

解释：
00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0

00 04：后面要发送的字节数
01:子站地址
02:指令号，02 读 DI
01:后面跟的数据字节数
00:换成二进制就是 0000 0000 =X8-X7-X6-X5-X4-X3-X2-X1 输入口的当前状态 0=OFF，1=ON

■ 例子：读 X1-X8 当前输入状态的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 02 00 00 00 08
解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
02:指令号，02 读 DI
00 00:从哪一路开始读，0x0000=X1 开始读
00 08:要读多少路十六进制 0x0008=十进制 8，读取 8 路

响应：

//X1,X8 输入 ON(1)时,其他都输入 OFF(0)时的回复
00 00 00 00 00 04 01 02 01 81
解释：
00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 04：后面要发送的字节数
01:子站地址
02:指令号，02 读 DI
01:后面跟的数据字节数
81:换成二进制就是 1000 0001 =X8-X7-X6-X5-X4-X3-X2-X1 输入口的当前状态 0=OFF，1=ON

■ 例子：读 X1-X24 当前输入状态的请求&响应报文

请求：

发送数据(HEX):00 00 00 00 00 06 01 02 00 00 00 18
解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
02:指令号，02 读 DI
00 00:从哪一路开始读，0x0000=X1 开始读
00 18:要读多少路十六进制 0x0018=十进制 24，读取 24 路

响应：

//X1,X2,X9,X24 输入 ON(1)时,其他都输入 OFF(0)时的回复

00 00 00 00 00 06 01 02 03 03 01 80

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

02:指令号，02 读 DI

03:后面跟的数据字节数

03:换成二进制就是 0000 0011 =X8-X7-X6-X5-X4-X3-X2-X1 输入口的当前状态 0=OFF，1=ON

01:换成二进制就是 0000 0001 =X16-X15-X14-X13-X12-X11-X10-X9 输入口的当前状态 0=OFF，1=ON

80:换成二进制就是 1000 0000 =X24-X23-X22-X21-X20-X19-X18-X17 输入口的当前状态 0=OFF，1=ON

2. X 输入口脉冲计数读取 (读取:AI 寄存器,命令号:0x04)

■ 描述

读模块的 X 输入口的脉冲计数值。

■ 例子：读当前 X1 输入点脉冲计数值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 04 00 18 00 02

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

00 18:从哪一路开始读，0x0018 对应的是 X1 脉冲计数寄存器起始地址

00 02:要读多少个寄存器，因为一路 X 脉冲计数占用 2 个寄存器地址，所以这里寄存器数量要填 2，十六进制 0x0002=十进制 2

响应：

接收数据(HEX): 00 00 00 00 00 07 01 04 04 00 00 27 10

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 07：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

04:后面跟的数据字节数,0x04=十进制 4，后面数据区有 4 字节数据

00002710:换成十进制就是 0x00002710 =10000,即读取到 X1 输入口当前脉冲计数值为 10000

■ 例子：读当前 X1-X4 输入点脉冲计数值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 04 00 18 00 08

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

00 18:从哪一路开始读，0x0018 对应的是 X1 脉冲计数寄存器起始地址

00 08:要读多少个寄存器，因为一路 X 脉冲计数占用 2 个寄存器地址，这里读 4 路，所以这里寄存器数量要填 8，十六进制 0x0008=十进制 8

响应：

接收数据(HEX): 00 00 00 00 00 13 01 04 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 13：后面要发送的字节数，0x13=十进制 19

01:子站地址

04:指令号，04 读 AI

10:后面跟的数据字节数,0x10=十进制 16，后面数据区有 16 字节数据

00002710:换成十进制就是 0x00002710 =10000,即读取到 X1 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X2 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X3 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X4 输入口当前脉冲计数值为 10000

■ 例子：读当前 X1-X8 输入点脉冲计数值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 04 00 18 00 10

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

00 18:从哪一路开始读，0x0018 对应的是 X1 脉冲计数寄存器起始地址

00 10:要读多少个寄存器，因为一路 X 脉冲计数占用 2 个寄存器地址，这里读 8 路，所以这里寄存器数量要填十六进制 0x0010=十进制 16

响应：

接收数据(HEX): 00 00 00 00 00 23 01 04 20 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 23：后面要发送的字节数，0x23=十进制 35

01:子站地址

04:指令号，04 读 AI

20:后面跟的数据字节数,0x20=十进制 32，后面数据区有 32 字节数据

00002710:换成十进制就是 0x00002710 =10000,即读取到 X1 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X2 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X3 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X4 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X5 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X6 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X7 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X8 输入口当前脉冲计数值为 10000

■ 例子：读当前 X1-X12 输入点脉冲计数值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 04 00 18 00 18

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

00 18:从哪一路开始读，0x0018 对应的是 X1 脉冲计数寄存器起始地址

00 18:要读多少个寄存器，因为一路 X 脉冲计数占用 2 个寄存器地址，这里读 12 路，所以这里寄存器数量要填十六进制 0x0018=十进制 24

响应：

接收数据(HEX): 00 00 00 00 00 33 01 04 30 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10 00 00 27 10

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 33：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

30:后面跟的数据字节数,0x30=十进制 48，后面数据区有 48 字节数据

00002710:换成十进制就是 0x00002710 =10000,即读取到 X1 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X2 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X3 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X4 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X5 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X6 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X7 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X8 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读取到 X9 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读到 X10 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读到 X11 输入口当前脉冲计数值为 10000

00002710:换成十进制就是 0x00002710 =10000,即读到 X12 输入口当前脉冲计数值为 10000

3. X 输入口脉冲计数清空 (写入:DO 寄存器,命令号:0x0F)

■ 描述

清空模块的 X 输入口的脉冲计数值。

■ 例子：清空 X1 输入计数值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 08 01 0f 00 40 00 01 01 01

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 08：后面要发送的字节数

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 01:要清空多少路计数，十六进制 0x0001=十进制 1 路

01:后面要写入的数据字节数,0x01=十进制 1，写入 1 字节数据

01:写入的数据，写 1 清空

响应：

//模块回复

00 00 00 00 00 06 01 0F 00 40 00 01

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 01:要清空多少路计数，十六进制 0x0001=十进制 1 路

■ 例子：清空 X1-X12 输入计数值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 09 01 0f 00 40 00 0C 02 ff 0f

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 09：后面要发送的字节数

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 0C:要清空多少路计数，十六进制 0x000C=十进制 12 路
02:后面要写入的数据字节数,0x02=十进制 2，写入 2 字节数据
ff:十六进制 0xff=二进制 1111 1111=X8 计数清空-X7 计数清空-X6 计数清空-X5 计数清空-X4 计数清空-X3 计数清空-X2 计数清空-X1 计数清空，对应的 bit 位写 1 清空，写 0 保持不变
0f: 十六进制 0x0f=二进制 0000 1111=X16 计数清空-X15 计数清空-X14 计数清空-X13 计数清空-X12 计数清空-X11 计数清空-X10 计数清空-X9 计数清空，对应的 bit 位写 1 清空，写 0 保持不变

响应：
//模块回复
00 00 00 00 00 06 01 0F 00 40 00 0C
解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址
00 0C:要清空多少路计数，十六进制 0x000C=十进制 12 路

■ 例子：指定清空 X1,X4,X8,X12,X24 输入计数值的请求&响应报文

请求：
发送数据(HEX): 00 00 00 00 00 0A 01 0f 00 40 00 18 03 85 08 80
解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 0A：后面要发送的字节数,0x0A=十进制 10
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址
00 18:要清空多少路计数，十六进制 0x0018=十进制 24 路
03:后面要写入的数据字节数,0x03=十进制 3，写入 3 字节数据
85:十六进制 0x85=二进制 1000 1001=X8 计数清空-X7 计数清空-X6 计数清空-X5 计数清空-X4 计数清空-X3 计数清空-X2 计数清空-X1 计数清空，对应的 bit 位写 1 清空，写 0 保持不变
08: 十六进制 0x08=二进制 0000 1000=X16 计数清空-X15 计数清空-X14 计数清空-X13 计数清空-X12 计数清空-X11 计数清空-X10 计数清空-X9 计数清空，对应的 bit 位写 1 清空，写 0 保持不变
80: 十六进制 0x80=二进制 1000 0000=X24 计数清空-X23 计数清空-X22 计数清空-X21 计数清空-X20 计数清空-X19 计数清空-X18 计数清空-X17 计数清空，对应的 bit 位写 1 清空，写 0 保持不变

响应：

//模块回复

00 00 00 00 00 06 01 0F 00 40 00 18

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 40:从哪一路开始清空，0x00 40 是 X1 计数清空 DO 寄存器起始地址

00 18:操作了多少路计数，十六进制 0x0018=十进制 24 路

4. Y 输出口 ON/OFF 写入 (写入:DO 寄存器,命令号:0x0F)

■ 描述

用于控制模块 Y 输出点的开关状态。

■ 例子：控制 Y1 当前输出 ON/OFF 的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 08 01 0f 00 00 00 01 01 01

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 08：后面要发送的字节数

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 00:从哪一路开始写，00 00=Y1 开始写

00 01:要写多少路，十六进制 0x0001=十进制 1，写 1 路

01:后面要写入的数据字节数,0x01=十进制 1，写入 1 字节数据

01:写入的数据，0x01 换算成二进制 00000001 = Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 输出状态，ON=0，OFF=1

响应：

//模块回复

00 00 00 00 00 06 01 0F 00 00 00 01

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 00:从哪一路开始写，00 00=Y1 开始写

00 01:写了多少路，十六进制 0x0001=十进制 1，写 1 路

■ 例子：控制 Y1-Y8 当前输出 ON/OFF 的请求&响应报文

//将 Y1，Y2，Y8 输出 ON，其他路输出 OFF

请求：

发送数据(HEX): 00 00 00 00 00 08 01 0f 00 00 00 08 01 83

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0
00 08：后面要发送的字节数
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 08:要写多少路，十六进制 0x0008=十进制 8，写 8 路
01:后面要写入的数据字节数,0x01=十进制 1，写入 1 字节数据
83:写入的数据，0x83 换算成二进制 10000011 = Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 输出状态，ON=0，OFF=1

响应：
//模块回复
00 00 00 00 00 06 01 0F 00 00 00 08
解释：
00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 08:写了多少路，十六进制 0x0008=十进制 8，写 8 路

■ 例子：控制 Y1-Y24 当前输出 ON/OFF 的请求&响应报文

//将 Y1，Y2，Y9，Y24 输出 ON，其他路输出 OFF
请求：
发送数据(HEX): 00 00 00 00 00 0a 01 0f 00 00 00 18 03 03 01 80
解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 0a：后面要发送的字节数,0x0a=十进制 10
01:子站地址
0f:指令号，0x0f=十进制 15，写多路 DO
00 00:从哪一路开始写，00 00=Y1 开始写
00 18:要写多少路，十六进制 0x0018=十进制 24，写 24 路
03:后面要写入的数据字节数,0x01=十进制 1，写入 1 字节数据
03:写入的数据，0x03 换算成二进制 00000011 = Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 输出状态，ON=0，OFF=1
01:写入的数据，0x01 换算成二进制 00000001 = Y16-Y15-Y14-Y13-Y12-Y11-Y10-Y9 输出状态，ON=0，OFF=1
80:写入的数据，0x80 换算成二进制 10000000 = Y24-Y23-Y22-Y21-Y20-Y19-Y18-Y17 输出状态，

ON=0，OFF=1

响应：

//模块回复

00 00 00 00 00 06 01 0F 00 00 00 18

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

0f:指令号，0x0f=十进制 15，写多路 DO

00 00:从哪一路开始写，00 00=Y1 开始写

00 18:写了多少路，十六进制 0x0018=十进制 24，写 24 路

5. Y 输出口 ON/OFF 读取 (读取:DO 寄存器,命令号:0x01)

■ 描述

用于读取模块 Y 输出点的开关状态。

■ 例子：读 Y1 当前输出状态的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 01 00 00 00 01

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

01:指令号，01 读取多路 DO

00 00:从哪一路开始读，00 00=Y1 开始读

00 01:要读多少路，十六进制 0x0001=十进制 1，读取 1 路

响应：

//Y1 输出 ON(1)时的回复

00 00 00 00 00 04 01 01 01 01

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 04：后面要发送的字节数

01:子站地址

01:指令号，01 读取多路 DO

01:后面数据区字节数

01:数据，换算成二进制 0x01=0000 0001,对应 Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 的状态

//Y1 输出 OFF(0)的时回复

00 00 00 00 00 04 01 01 01 00

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 04：后面要发送的字节数

01:子站地址

01:指令号，01 读取多路 DO

01:后面数据区字节数

00:数据，换算成二进制 0x00=0000 0000,对应 Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 的状态

■ 例子：读 Y1-Y8 当前输出状态的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 01 00 00 00 08

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

01:指令号，01 读取多路 DO

00 00:从哪一路开始读，00 00=Y1 开始读

00 08:要读多少路，十六进制 0x0008=十进制 8，读取 8 路

响应：

//Y1,Y2,Y8 输出 ON(1)，其他输出 OFF(0)时的回复

00 00 00 00 00 04 01 01 01 83

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 04：后面要发送的字节数

01:子站地址

01:指令号，01 读取多路 DO

01:后面数据区字节数,0x01=十进制 1，后面数据区有 1 字节数据

83:数据区，换算成二进制 0x83=1000 0011,对应 Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 的状态

■ 例子：读 Y1-Y24 当前输出状态的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 01 00 00 00 18

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

01:指令号，01 读取多路 DO

00 00:从哪一路开始读，00 00=Y1 开始读

00 18:要读多少路，十六进制 0x0018=十进制 24，读取 24 路

响应：

//Y1,Y2,Y9,Y24 输出 ON(1)，其他输出 OFF(0)时的回复

00 00 00 00 00 06 01 01 03 03 01 80

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址
01:指令号，01 读取多路 DO
03:后面数据区字节数,0x03=十进制 3，后面数据区有 3 字节数据
03:数据，换算成二进制 0x03=0000 0011,对应 Y8-Y7-Y6-Y5-Y4-Y3-Y2-Y1 的状态
01:数据，换算成二进制 0x01=0000 0001,对应 Y16-Y15-Y14-Y13-Y12-Y11-Y10-Y9 的状态
80:数据，换算成二进制 0x80=1000 0000,对应 Y24-Y23-Y22-Y21-Y20-Y19-Y18-Y17 的状态

6. AI 模拟量采集读取(读取:AI 寄存器,命令号:0x04)

■ 描述

读模块中的输入型数据,例如 AD 模拟采集到的电压值,电流值,压力值等。

■ 例子：读当前 AI 通道 AI1 模拟量输入值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 04 00 00 00 02

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

00 00:从哪一路开始读，0x0000=AIO1 开始读

00 02:要多少个寄存器,因为一个 AI 通道占用 2 个寄存器地址,所以这里读 1 路模拟量的话，寄存器数量要填 2，十六进制 0x0002=十进制 2

响应：

接收数据(HEX): 00 00 00 00 00 07 01 04 04 00 00 BF 11

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 07：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

04:后面跟的数据字节数,0x04=十进制 4，后面数据区有 4 字节数据

0000BF11:换成十进制就是 0x0000BF11 =48913,即读取到 AI1 模拟量输入口当前值为 48913

■ 例子：读当前 AI 通道 AI1-AI4 模拟量输入值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 04 00 00 00 08

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

00 00:从哪一路开始读，0x0000=AIO1 开始读

00 08:要多少个寄存器，因为一个 AI 通道占用 2 个寄存器地址，所以这里读 4 路模拟量的话，寄存器数量要填 8，十六进制 0x0008=十进制 8

响应：

接收数据(HEX): 00 00 00 00 00 13 01 04 10 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 13：后面要发送的字节数，0x13=十进制 19

01:子站地址

04:指令号，04 读 AI

10:后面跟的数据字节数,0x10=十进制 16，后面数据区有 16 字节数据

0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI1 模拟量输入口当前值为 48913

0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI2 模拟量输入口当前值为 48913

0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI3 模拟量输入口当前值为 48913

0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI4 模拟量输入口当前值为 48913

■ 例子：读当前 AI 通道 AI1-AI8 模拟量输入值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 04 00 00 00 10

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

04:指令号，04 读 AI

00 00:从哪一路开始读，0x0000=AIO1 开始读

00 10:要多少个寄存器，因为一个 AI 通道占用 2 个寄存器地址，所以这里读 8 路模拟量的话，寄存器数量要填 16，换算成十六进制就是 0x0010

响应：

接收数据(HEX): 00 00 00 00 00 23 01 04 20 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11 00 00 BF 11

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 23：后面要发送的字节数，0x23=十进制 35

01:子站地址

04:指令号，04 读 AI

20:后面跟的数据字节数,0x20=十进制 32，后面数据区有 32 字节数据

0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI1 模拟量输入口当前值

为 48913
0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI2 模拟量输入口当前值为 48913
0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI3 模拟量输入口当前值为 48913
0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI4 模拟量输入口当前值为 48913
0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI5 模拟量输入口当前值为 48913
0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI6 模拟量输入口当前值为 48913
0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI7 模拟量输入口当前值为 48913
0000BF11:数据区，换成十进制就是 0x0000BF11 =48913,即读取到 AI8 模拟量输入口当前值为 48913

7. AO 模拟量输出写入(写入:AO 寄存器,命令号:0x10)

■ 描述

用于设定 DA 模拟量输出值。

■ 例子：写当前 AO 通道 AO1 模拟量输出值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 09 01 10 00 00 00 01 02 10 D2

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 09：后面要发送的字节数
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 01:要写多少路，十六进制 0x0001=十进制 1，写 1 路
02:后面要写入的数据字节数，0x02=十进制 2，写 2 个字节
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO1 输出口模拟量输出值为 4306

响应：

接收数据(HEX): 00 00 00 00 00 06 01 10 00 00 00 01

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 01:写了多少路，十六进制 0x0001=十进制 1，写 1 路

■ 例子：写当前 AO 通道 AO1-AO4 模拟量输出值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 0f 01 10 00 00 00 04 08 10 D2 10 D2 10 D2 10 D2

解释：

00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 0f：后面要发送的字节数,0x0f=十进制 15

01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 04:要写多少路，十六进制 0x0004=十进制 4，写 4 路
08:后面要写入的数据字节数，0x08=十进制 8，写 8 个字节
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO1 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO2 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO3 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO4 输出口模拟量输出值为 4306

响应：
接收数据(HEX): 00 00 00 00 00 06 01 10 00 00 00 04
解释：
00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 04:写了多少路，十六进制 0x0004=十进制 4，写 4 路

■ 例子：写当前 AO 通道 AO1-AO8 模拟量输出值的请求&响应报文

请求：
发送数据(HEX): 00 00 00 00 00 17 01 10 00 00 00 08 10 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2
解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 17：后面要发送的字节数，0x17=十进制 23
01:子站地址
10:指令号，0x10=十进制 16，写 AO
00 00:从哪一路开始写，0x0000=AIO1 开始写
00 08:要写多少路，十六进制 0x0008=十进制 8，写 8 路
10:后面要写入的数据字节数，0x10=十进制 16，16 个字节
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO1 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO2 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO3 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO4 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO5 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO6 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO7 输出口模拟量输出值为 4306
10D2:要写入的 AO 输出值，0x10D2=十进制 4306,即写入 AO8 输出口模拟量输出值为 4306

响应：

接收数据(HEX): 00 00 00 00 00 06 01 10 00 00 00 08

解释：

00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0

00 00：PID 协议标识符，默认 0

00 06：后面要发送的字节数

01:子站地址

10:指令号，0x10=十进制 16，写 AO

00 00:从哪一路开始写，0x0000=AIO1 开始写

00 08:写了多少路，十六进制 0x0008=十进制 8，写 8 路

8. AO 模拟量输出读取(读取:AO 寄存器,命令号:0x03)

■ 描述

用于读取当前 DA 模拟量输出值。

■ 例子：读当前 AO 通道 AO1 模拟量输出值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 03 00 00 00 01

解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
03:指令号，03 读 AO
00 00:从哪一路开始读，0x0000=AIO1 开始读
00 01:要读多少路，十六进制 0x0001=十进制 1，读取 1 路

响应：

接收数据(HEX): 00 00 00 00 00 05 01 03 02 10 D2

解释：
00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 05：后面要发送的字节数
01:子站地址
03:指令号，03 读 AO
02:后面跟的数据字节数
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO1 输出模拟量当前输出值为 4306

■ 例子：读当前 AO 通道 AO1-AO4 模拟量输出值的请求&响应报文

请求：

发送数据(HEX): 00 00 00 00 00 06 01 03 00 00 00 04

解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
03:指令号，03 读 AO

00 00:从哪一路开始读，0x0000=AIO1 开始读
00 04:要读多少路，十六进制 0x0004=十进制 4，读取 4 路

响应：
接收数据(HEX): 00 00 00 00 00 0B 01 03 08 10 D2 10 D2 10 D2 10 D2

解释：
00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 0B：后面要发送的字节数,0x000B=十进制 11
01:子站地址
03:指令号，03 读 AO
08:后面跟的数据字节数
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO1 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO2 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO3 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO4 输出模拟量当前输出值为 4306

■ 例子：读当前 AO 通道 AO1-AO8 模拟量输出值的请求&响应报文

请求：
发送数据(HEX): 00 00 00 00 00 06 01 03 00 00 00 08

解释：
00 00：TID 传输标识符(用于上位机传输报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 06：后面要发送的字节数
01:子站地址
03:指令号，03 读 AO
00 00:从哪一路开始读，0x0000=AIO1 开始读
00 08:要读多少路，十六进制 0x0008=十进制 8，读取 8 路

响应：
接收数据(HEX): 00 00 00 00 00 13 01 03 10 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2 10 D2

解释：
00 00：TID 传输标识符(用于模块回送报文序列号)，也可为 0
00 00：PID 协议标识符，默认 0
00 13：后面要发送的字节数，0x0013=十进制 19
01:子站地址
03:指令号，03 读 AO

10:后面跟的数据字节数,十六进制 0x10=十进制 16
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO1 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO2 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO3 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO4 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO5 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO6 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO7 输出模拟量当前输出值为 4306
10 D2:换成十进制就是 0x10D2 =十进制 4306,即读取到 AO8 输出模拟量当前输出值为 4306

谢

谢！