

IC2004DO 四回路控制器

使用说明书

2022年2月

目录

1. 产品概述.....	1
1.1. 概述.....	1
1.2. 性能特点.....	1
1.3. 技术参数.....	2
2. 外观尺寸.....	4
3. 产品接线图.....	6
4. 引脚说明及指示灯.....	7
4.1. 引脚定义.....	7
4.2. LED 指示灯.....	7
5. 通讯协议及寄存器定义.....	9
5.1. 通讯协议.....	9
5.2. 读线圈状态.....	9
5.3. 写单个线圈状态.....	10
5.4. 写多个线圈状态.....	11
5.5. 读保持寄存器.....	13
5.6. 写单个保持寄存器.....	14
5.7. 写多个保持寄存器.....	15
5.8. 错误码表.....	16
5.9. 寄存器定义.....	16
5.10. 协议应用范例.....	21

1. 产品概述

1.1. 概述

IC2004 为智能数字量采集器，具有 4 路干接点数字量输入（DI）和 4 路 C 型继电器输出数字量（DO）；电源及 RS485 接口均加入防雷保护电路，产品稳定可靠；丰富的指示灯方便调试，运行状态一目了然；采用标准 Modbus RTU 协议，符合工业标准，方便系统集成商、工程商使用；方便与上位机通讯，可实现快速组网，构建监测系统；适用于各种工业场合及自动化系统。通过 RS-485 即可实现对远程数字量设备的数据采集和控制。

本产品采用标准 Modbus RTU 协议及常用功能码，使用户可以更加轻松实现与 SCADA 软件、HMI 设备及支持 Modbus RTU 协议的 PLC 等设备和系统的整合应用；提供协议和示例代码，使您的二次开发更加灵活、简便、高效。

1.2. 性能特点

- 4 路干接点数字量输入（DI）
- 4 路 C 型继电器输出（DO）
- DI 输入范围：0~5V
- DO 输出：3A，250VAC
- 双硬件看门狗，绝不死机
- 采用 32 位 ARM 嵌入式 CPU，高性能低功耗
- 采用 Modbus RTU 通信协议
- 丰富的的指示灯，方便调试
- RS485 通信接口提供光电隔离及防雷保护
- 电源具有过流、过压、防反接及防雷保护

- 工业级温度范围，应对严苛现场环境
- 标准导轨安装或螺钉固定

1.3. 技术参数

DI 接口	DI	4 路干接点
	接点电压	小于 1V 为逻辑 1，大于 2.5V 为逻辑 0
	触发电流	小于 1mA
	防雷防护	共模，600W
	过压过流保护	30V，500mA
DO 接口	DO	4 路 C 型继电器
	接点电压	0~250VAC
	触点容量	3A
串口通信参数	通讯接口	RS485
	波特率	1200~115200bps
	数据格式	N.8.1
	通讯协议	Modbus RTU
	防雷防护	共模，600W
	过压过流保护	30V，100mA
电源参数	电源规格	9-28VDC(推荐 12VDC)
	功耗	60mA@12VDC
	防雷防护	1300W

	过压过流保护	30V, 500mA
工作环境	工作温度、湿度	-40~85℃, 5~90%RH, 不凝露
	储存温度、湿度	-60~125℃, 5~90%RH, 不凝露
其他	尺寸	110mm*75mm*30mm
	保修	1年质保

2. 外观尺寸



图-产品外观

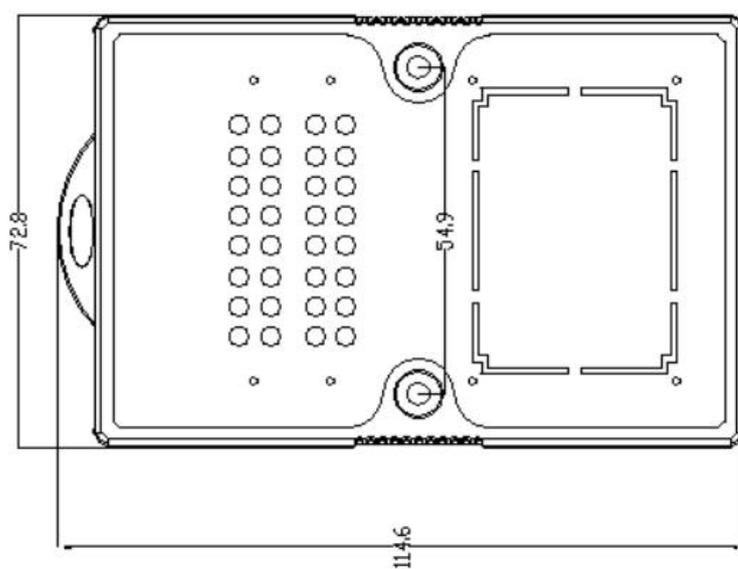


图-前视图

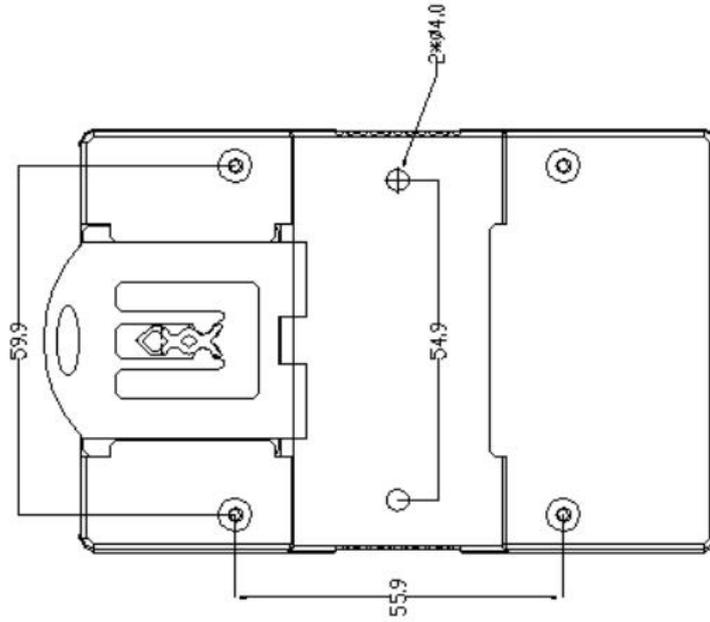


图-后视图

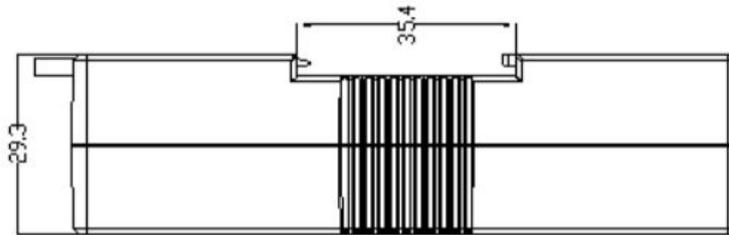


图-侧视图

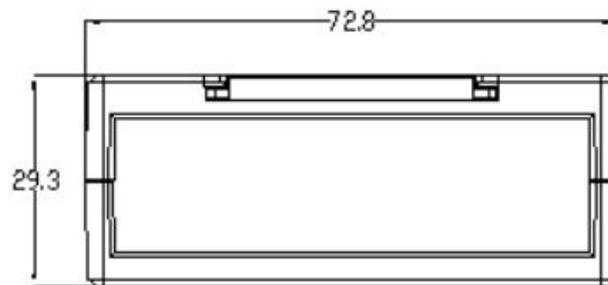
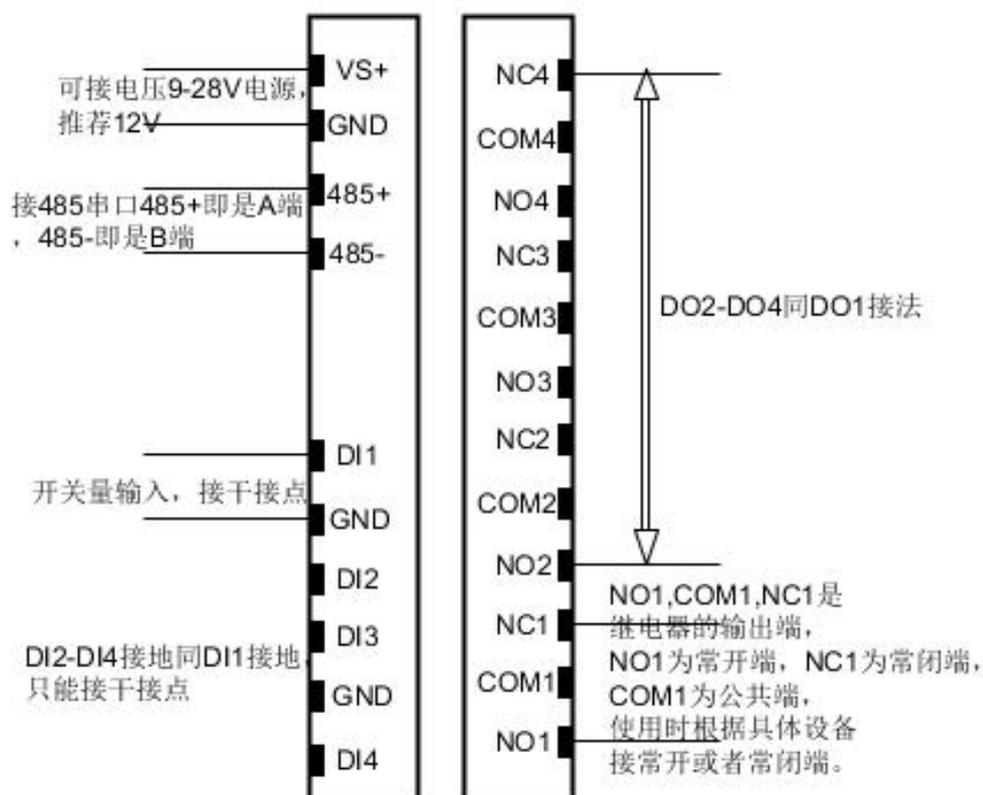


图-顶视图

3. 产品接线图



4. 引脚说明及指示灯

4.1. 引脚定义

引脚定义	说明
VS+	电源正
GND	电源负
485+	RS485+
485-	RS485-
DI(GND)	数字量信号输入公共端
DI1~4	数字量信号输入端
DO1~4(COM)	数字量信号输出端
DO1~4(NC)	数字量信号常闭输出端
DO1~4(NO)	数字量信号常开输出端空

4.2. LED 指示灯

IC2004 外设 12 个状态 LED 指示灯,能够准确及时报告设备的工作状态,为工程的施工和调试带来极大的方便。其说明如下表所示:

指示灯	指示灯说明
PWR	电源指示灯 (亮: 有电源连接; 灭: 无电源连接)
RUN	闪烁: 正常运行; 常亮或者不亮: 工作不正常
TX	RS485 接口发送数据

RX	RS485 接口接收数据
DI1~DI4	亮：对应 DI 有输入
DO1~DO4	亮：对应 DO 有输出

5. 通讯协议及寄存器定义

5.1. 通讯协议

遵循标准 MODBUS RTU 协议，协议格式如下：

从设备地址	功能码	数据	校验
1 字节	1 字节	N 字节	2 字节

从设备地址：即 IO 模块的地址，地址可设置；

功能码：读写 IO 模块 DIO 状态的功能码；

数据：根据功能码和寄存器个数确定数据的大小；

校验：CRC16 校验，校验低位在前，高位在后。

5.2. 读线圈状态

功能码：0x01

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x01
起始寄存器地址	2 字节，高位在前
寄存器个数	2 字节，高位在前
CRC16 校验	2 字节，低位在前

IO 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
-------	--------------------

功能码	1 字节，内容为 0x01
字节数	1 字节，从读寄存器个数计算得出： 如果寄存器个数被 8 整除： 字节数=寄存器个数/8 如果寄存器个数不能被 8 整除： 字节数=寄存器个数/8+1
数据	每一位表示一路 DIO 的状态，第一个字节的第二位表示起始寄存器的状态，依次类推
CRC16 校验	2 字节，低位在前

IO 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x01
数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

5.3. 写单个线圈状态

功能码：0x05

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x05
寄存器	2 字节，高位在前

寄存器值	2 字节，高位在前，写 0x0000 表示输出 0，写 0xff00 表示输出 1
CRC16 校验	2 字节，低位在前

I0 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x05
寄存器	2 字节，高位在前
寄存器值	2 字节，高位在前，回应 0x0000 表示 0，回应 0xff00 表示 1
CRC16 校验	2 字节，低位在前

I0 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x05
数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

5.4. 写多个线圈状态

功能码：0x0f

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x0f

起始寄存器	2 字节，高位在前
寄存器个数	2 字节，高位在前
字节数	1 字节，字节数从寄存器个数计算得出： 如果寄存器个数被 8 整除： 字节数=寄存器个数/8 如果寄存器个数不能被 8 整除： 字节数=寄存器个数/8+1
数据	每一位表示一路线圈状态（即是 DO 或其配置），第一个字节的第 1 位表示起始寄存器的状态
CRC16 校验	2 字节，低位在前

I0 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x0f
起始寄存器	2 字节，高位在前
寄存器个数	2 字节，高位在前
CRC16 校验	2 字节，低位在前

I0 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x0f
数据	1 字节，错误码，见错误码表

CRC16 校验	2 字节，低位在前
----------	-----------

5.5. 读保持寄存器

功能码：0x03

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x03
起始寄存器地址	2 字节，高位在前
寄存器个数	2 字节，高位在前
CRC16 校验	2 字节，低位在前

I0 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x03
字节数	1 字节，即是寄存器个数 x2，因为每个保持寄存器两个字节
数据	各个保持寄存器的值，每个保持寄存器占用 2 字节，并且高位在前
CRC16 校验	2 字节，低位在前

I0 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x03

数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

5.6. 写单个保持寄存器

功能码：0x06

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x06
寄存器	2 字节，高位在前
寄存器值	2 字节，高位在前
CRC16 校验	2 字节，低位在前

I0 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x06
寄存器	2 字节，高位在前
寄存器值	2 字节，高位在前
CRC16 校验	2 字节，低位在前

I0 模块异常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x80+0x06

数据	1 字节，错误码，见错误码表
CRC16 校验	2 字节，低位在前

5.7. 写多个保持寄存器

功能码：0x10

上位机报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x10
起始寄存器地址	2 字节，高位在前
寄存器个数	2 字节，高位在前
字节数	1 字节，即是寄存器个数 x2，因为每个保持寄存器占用 2 个字节
数据	各个保持寄存器的值，每个保持寄存器占用 2 字节，并且高位在前
CRC16 校验	2 字节，低位在前

I0 模块正常应答报文：

从设备地址	1 字节，内容为 0x00-0xff
功能码	1 字节，内容为 0x10
起始寄存器地址	2 字节，高位在前
寄存器个数	2 字节，高位在前
CRC16 校验	2 字节，低位在前

I0 模块异常应答报文:

从设备地址	1 字节, 内容为 0x00-0xff
功能码	1 字节, 内容为 0x80+0x10
数据	1 字节, 错误码, 见错误码表
CRC16 校验	2 字节, 低位在前

5.8. 错误码表

错误码	意义
0x01	无效功能码
0x02	无效寄存器地址
0x03	寄存器值无效
0x04	从机设置错误
0x05	ACK, 一般用于长时间执行某项任务
0x06	从机忙状态
0x07	NEGATIVE ACK
0x08	MEMORY PARITY ERROR

5.9. 寄存器定义

5.9.1. 公共寄存器

寄存器地址	功能	种类	读写状态	取值范围
-------	----	----	------	------

0	型号高位	保持寄存器	只读	
1	型号低位	保持寄存器	只读	
2	序列号高位	保持寄存器	只读	
3	序列号低位	保持寄存器	只读	
4	版本高位	保持寄存器	只读	
5	版本低位	保持寄存器	只读	
6	BOOT 版本高位	保持寄存器	只读	
7	BOOT 版本低位	保持寄存器	只读	
8	波特率高位	保持寄存器	读写	300~115200
9	波特率低位	保持寄存器	读写	
10	地址	保持寄存器	读写	0~255
11	型号名字 1	保持寄存器	只读	字符串格式的型号名字，例如： IC2004
12	型号名字 2	保持寄存器	只读	
13	型号名字 3	保持寄存器	只读	
14	型号名字 4	保持寄存器	只读	
15	型号名字 5	保持寄存器	只读	
16	IO 模块寄存器基址	保持寄存器	只读	
17	IO 模块寄存器个数	保持寄存器	只读	
18	IO 模块配置总个数	保持寄存器	只读	

19	寄存器个数 1	保持寄存器	只读	某种类型的寄存器个数
20	寄存器类型 1	保持寄存器	只读	类型取值为: 0 保持寄存器, 1 线圈寄存器
21	寄存器属性 1	保持寄存器	只读	属性按位表示:1 可读, 2 可写, 3 可配置, 4 电平型 DI, 5 脉冲型 DI
22	是否浮点数 1	保持寄存器	只读	
.....		保持寄存器	只读	
22+4N	寄存器个数 N	保持寄存器	只读	某种类型的寄存器个数
23+4N	寄存器类型 N	保持寄存器	只读	类型取值为: 0 保持寄存器, 1 线圈寄存器
24+4N	寄存器属性 N	保持寄存器	只读	属性按位表示:1 可读, 2 可写, 3 可配置, 4 电平型 DI, 5 脉冲型 DI
25+4N	是否浮点数 N	保持寄存器	只读	

5.9.2. IC2004 寄存器

寄存器地址	功能	种类	读写状态	取值范围
100	DI 电平输入 1	线圈状态	只读	0 表示无输入, 1 表示有输入
101	DI 电平输入 2	线圈状态	只读	0 表示无输入, 1 表示有输入
102	DI 电平输入 3	线圈状态	只读	0 表示无输入, 1 表示有输入
103	DI 电平输入 4	线圈状态	只读	0 表示无输入, 1 表示有输入

104	DI 脉冲输入 1	线圈状态	只读	0 表示 DI1 无跳变, 1 表示 DI1 从 0 到 1 的跳变
105	DI 脉冲输入 2	线圈状态	只读	0 表示 DI2 无跳变, 1 表示 DI2 从 0 到 1 的跳变
106	DI 脉冲输入 3	线圈状态	只读	0 表示 DI3 无跳变, 1 表示 DI3 从 0 到 1 的跳变
107	DI 脉冲输入 4	线圈状态	只读	0 表示 DI4 无跳变, 1 表示 DI4 从 0 到 1 的跳变
108	DO 输出 1	线圈状态	读写	0 表示无输出, 1 表示有输出
109	DO 输出 2	线圈状态	读写	0 表示无输出, 1 表示有输出
110	DO 输出 3	线圈状态	读写	0 表示无输出, 1 表示有输出
111	DO 输出 4	线圈状态	读写	0 表示无输出, 1 表示有输出
112	上电 DO 配置 1	线圈状态	读写	0 表示无输出, 1 表示有输出
113	上电 DO 配置 2	线圈状态	读写	0 表示无输出, 1 表示有输出
114	上电 DO 配置 3	线圈状态	读写	0 表示无输出, 1 表示有输出
115	上电 DO 配置 4	线圈状态	读写	0 表示无输出, 1 表示有输出
100	DI 电平输入 1	保持寄存器	只读	0 表示无输入, 1 表示有输入
101	DI 电平输入 2	保持寄存器	只读	0 表示无输入, 1 表示有输入
102	DI 电平输入 3	保持寄存器	只读	0 表示无输入, 1 表示有输入
103	DI 电平输入 4	保持寄存器	只读	0 表示无输入, 1 表示有输入
104	DI 脉冲输入 1	保持寄存器	只读	0 表示 DI1 无跳变, 1 表示 DI1 从 0 到 1

				的跳变
105	DI 脉冲输入 2	保持寄存器	只读	0 表示 DI2 无跳变, 1 表示 DI2 从 0 到 1 的跳变
106	DI 脉冲输入 3	保持寄存器	只读	0 表示 DI3 无跳变, 1 表示 DI3 从 0 到 1 的跳变
107	DI 脉冲输入 4	保持寄存器	只读	0 表示 DI4 无跳变, 1 表示 DI4 从 0 到 1 的跳变
108	DO 输出 1	保持寄存器	读写	0 表示无输出, 1 表示有输出
109	DO 输出 2	保持寄存器	读写	0 表示无输出, 1 表示有输出
110	DO 输出 3	保持寄存器	读写	0 表示无输出, 1 表示有输出
111	DO 输出 4	保持寄存器	读写	0 表示无输出, 1 表示有输出
112	上电 DO 配置 1	保持寄存器	读写	0 表示无输出, 1 表示有输出
113	上电 DO 配置 2	保持寄存器	读写	0 表示无输出, 1 表示有输出
114	上电 DO 配置 3	保持寄存器	读写	0 表示无输出, 1 表示有输出
115	上电 DO 配置 4	保持寄存器	读写	0 表示无输出, 1 表示有输出
116	DO1 输出脉冲时间	保持寄存器	读写	0~65535ms
117	DO2 输出脉冲时间	保持寄存器	读写	0~65535ms
118	DO3 输出脉冲时间	保持寄存器	读写	0~65535ms
119	DO4 输出脉冲时间	保持寄存器	读写	0~65535ms

5.10. 协议应用范例

5.10.1. 读寄存器命令举例

以下为读取 I0 模块 4 路电平型 DI 和 4 路脉冲型 DI 的命令举例，假定 I0 模块的地址为 1，寄存器起始地址为 100（十六进制为 0x64），个数为 8，上位机发送的数据如下（十六进制表示）：

01 01 00 64 00 08 7c 13

各项分别表示：

01 I0 模块的地址，1 字节；

01 功能码：读取线圈状态的功能码；

00 64 起始寄存器，即是寄存器 100；

00 08 需要读取的寄存器个数，这里举例为 8 路，4 路电平型 DI 和 4 路脉冲型 DI；

7c 13 CRC16 校验，从地址到数据域的校验，计算结果为 0x137c，因为要低在前，所以是 7c 13。

从机应答举例，假定 4 路电平 DI 状态状态分别：1 0 1 0，脉冲型 DI 状态是电平型 DI 状态的脉冲表示，瞬间值为：1 0 1 0，则回应的数据如下（十六进制表示）：

01 01 01 55 91 b7

各项分别表示：

01 I0 模块的地址，1 字节；

01 功能码：读取线圈状态的功能码；

01 字节数，因为是 8 个寄存器，所以字节数=寄存器个数/8=1；

55 各个寄存器的值，从低位开始对应的电平 DI 的第一路；

91 b7 CRC16 校验，从地址到数据域的校验，计算结果为 0xb791，因为要低位在前，所以是 91 b7。

5.10.2. 写单个寄存器命令举例

以下为写 D01 输出的应用举例，假定 IO 模块的地址为 1，寄存器地址为 108（十六进制为 0x6c），写 D01 状态 1 的数据如下（十六进制表示）：

01 05 00 6c ff 00 4c 27

01 IO 模块的地址，1 字节；

05 功能码：写线圈状态的功能码；

00 6c 寄存器地址，高位在前，D01 的寄存器；

ff 00 向 D01 写 1 的操作，如果写 0，则填 00 00；

4c 27 CRC16 校验，从地址到数据域的校验，计算结果为 0x274c，因为要低位在前，所以是 4c 27。

如果执行正常，从机应答数据如下（十六进制表示）：

01 05 00 6c ff 00 4c 27

01 IO 模块的地址，1 字节；

05 功能码：写线圈状态的功能码；

00 6c 寄存器地址，高位在前，D01 的寄存器；

ff 00 D01 的状态返回值，ff 00 表示 D01 状态为 1，00 00 表示 D01

状态为 0;

4c 27 CRC16 校验，从地址到数据域的校验，计算结果为 0x274c，因为要低位在前，所以是 4c 27。

5.10.3. 写多个寄存器命令举例

以下为写从 D01 开始的 4 路 D0，假定 IO 模块的地址为 1，寄存器地址为 108（十六进制为 0x6c），写从 D01 到 D04 的数据（全部输出）如下（十六进制表示）：

01 0f 00 6c 00 04 01 0f ee 9b

01 IO 模块的地址，1 字节；

0f 功能码：写多路线圈状态的功能码；

00 6c 寄存器地址，高位在前，从 D01 寄存器开始写；

00 04 寄存器个数，写 4 路，即是 D01~D04；

01 字节数，寄存器个数不能被 8 整除，所以字节数=寄存器个数/8+1=1；

0f 各路 D0 寄存器的值，该字节的第一位表示第一路 D0，以此类推；

ee 9b CRC16 校验，从地址到数据域的校验，计算结果为 0x9bee，因为要低位在前，所以是 ee 9b。

如果执行正常，从机应答数据如下（十六进制表示）：

01 0f 00 6c 00 04 94 15

01 IO 模块的地址，1 字节；

0f 功能码：写多路线圈状态的功能码；

00 6c 寄存器地址，高位在前，从 D01 寄存器开始写；

00 04 寄存器个数，总共写 4 路，即是 D01~D04；

94 15 CRC16 校验，从地址到数据域的校验，计算结果为 0x1594，因为要低位在前，所以是 94 15。