



鹏合电子

PENGHE ELECTRONICS

工业级模拟量采集模块 产品说明书

PRODUCT DESCRIPTION



目录

一、产品概述	1
1.1 概述	1
1.2 性能特点	1
1.3 技术参数	1
二、外观尺寸	2
2.1 产品外观	2
2.2 产品尺寸图	3
三、产品接线图、跳线、指示灯说明	3
3.1 接线图	3
VDC GND IN1 IN2 IN3 IN4	3
有源传感器	3
3.2 跳线	4
3.3 LED 指示灯	5
四、软件操作	5
4.1 配置软件	5
4.2 配置基本参数	6
4.3 DO 输出相关参数	6
4.4 AI 模拟量采集相关参数	7
4.5 注意事项	8
五、ModbusRTU 通讯协议、组态软件软件说明	9
5.1 通讯协议	9
5.2 寄存器地址	9
5.2 Modbus RTU 功能码	9
5.3 Modbus 通讯实例	10
六、协议详解	10
6.1 功能码描述	11
6.1.1 01 读线圈	11
发送：640100140002F43A DTU 响应：64010101C434	11
6.1.2 03 读保持寄存器	12
6.1.3 04 读输入寄存器	12
发送：6403000000184C35	12
6.1.4 05 写单个线圈	13
发送：64050015FF00940B DTU 响应：64050015FF00940B	13
6.1.5 06 写单个寄存器	14
6.1.6 0F 写多个线圈	15
发送：640F001400020103A942 DTU 响应：640F001400029DFB	15
6.1.7 10 写多个寄存器	16
发送：64100014000204000100004D5D DTU 响应：6410001400020839	17
6.2 错误码描述	18
6.3 CRC 校验算法	18

一、产品概述

1.1 概述

该产品一款工业级标准模拟量采集产品，共有 4 个测量通道。每个通道均可以分别设置多种量程；RS-485 通讯接口使用标准 MODBUS RTU 协议，符合工业标准。

1.2 性能特点

- 防死机硬件看门狗
- 5~40V 带防反接、过压过流保护电源
- 4 路模拟量电流输入 0~20MA
- 4 路模拟量电压输入 0~5V、0~30V
- 12 位分辨率，0.1%精度 ADC
- 高性能低功耗 32 位 ARM 嵌入式 CPU
- 支持 MODBUSRTU 从站协议
- 2 路输出状态指示灯
- 带防雷、静电保护 RS485 通讯接口
- 工业机温度范围，应对严苛现场环境
- 自定义线性模拟量数据转换

1.3 技术参数

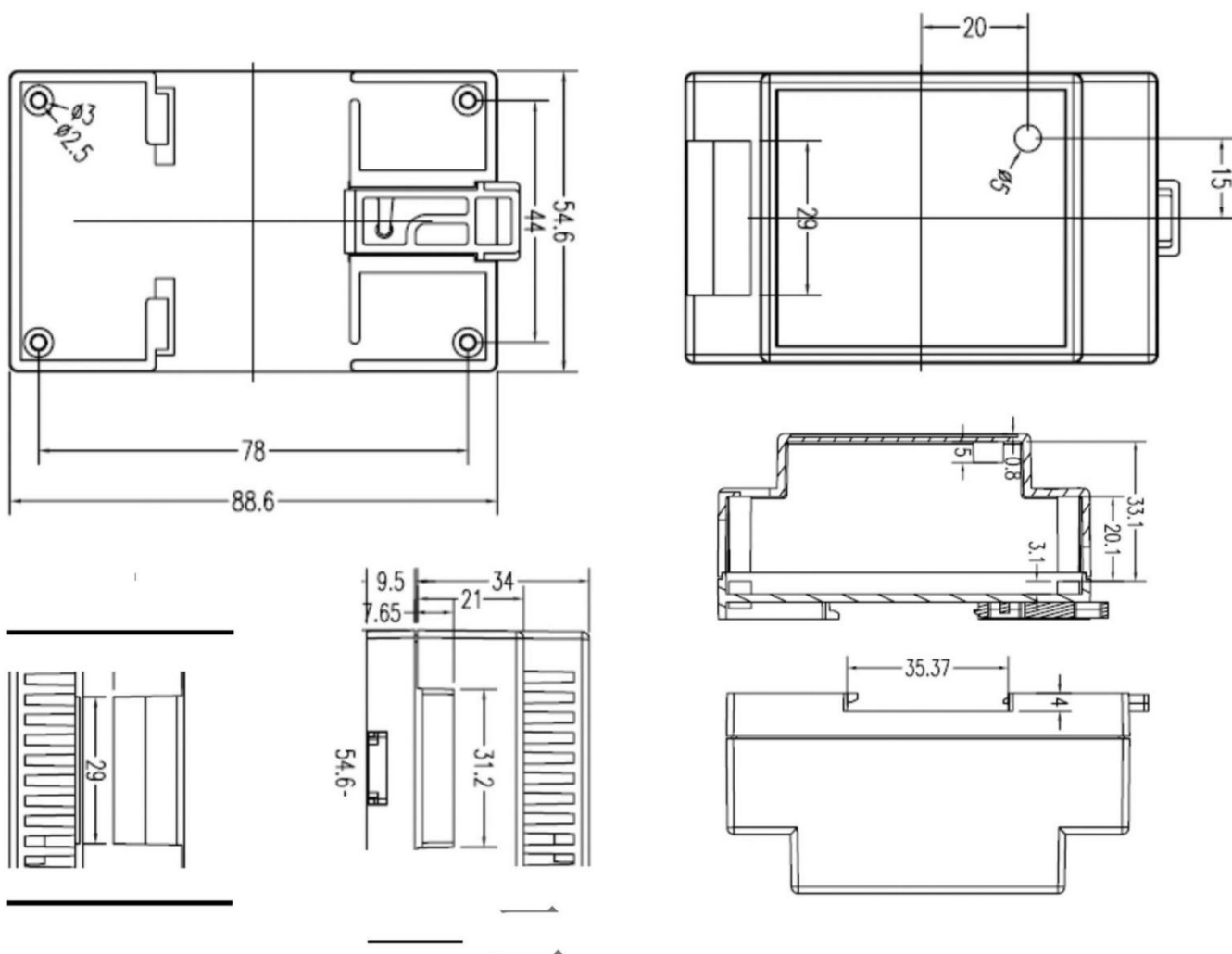
模拟量接口	AI	4路单端
	AI分辨率	12BIT
	AI量程	0~5V、0~30V、0~20MA
	精度	0.1%
	采集速度	1KHZ
	AI输入阻抗	0~20MA \leq 120 Ω 0~5V/0~30V \geq 10K Ω
开关量接口	DO	2 路单端
	输出信号类型	NPN
	输出电流电压	VOUT=VCC IM=1A
通讯接口	通讯接口	RS485
	波特率	1200~115200BPS
	数据格式	N.8.1
	通讯协议	MODBUSRTU
	过压过流保护	45V
电源参数	电源规格	5~35V
	功耗	12V-25MA
工作环境	工作温度、湿度	-40°C~85°C, 0%RH~95%RH
其他	尺寸	82*50*32

二、外观尺寸

2.1 产品外观

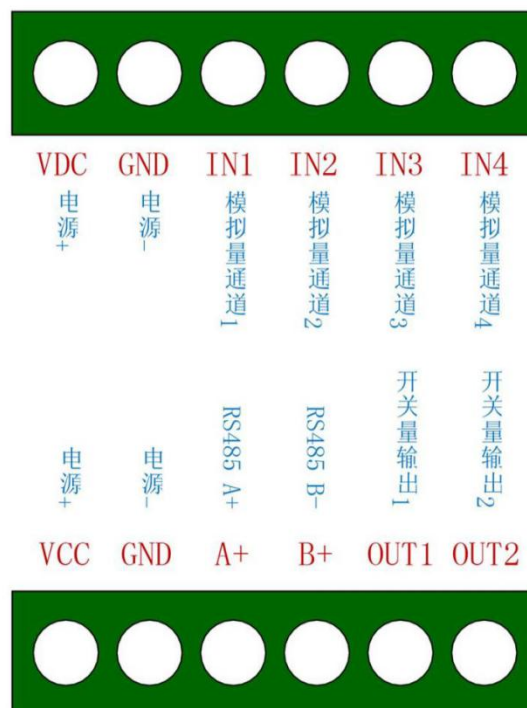
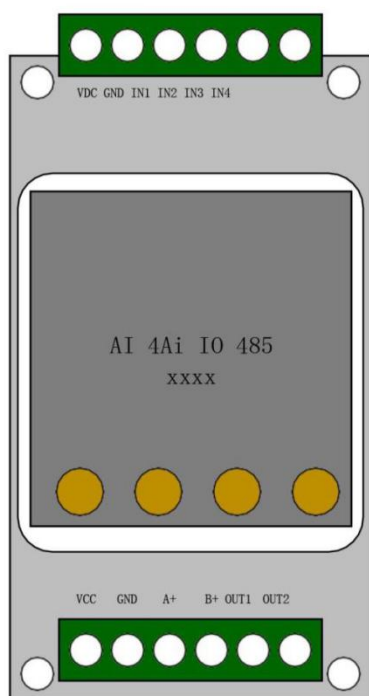


2.2 产品尺寸图



三、产品接线图、跳线、指示灯说明

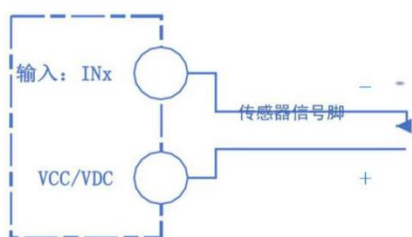
3.1 接线图



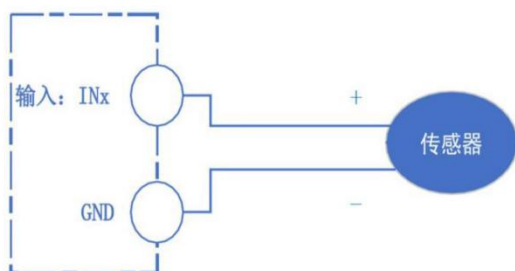
顶部 6 槽接线位:

VDC:	电源正极端	IN1: 模拟量输入端 1	IN2: 模拟量输入端 2
GND:	电源负极端	IN3: 模拟量输入端 3	IN4: 模拟量输入端 4

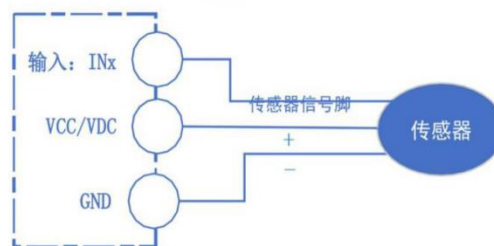
两线制传感器（无源）:



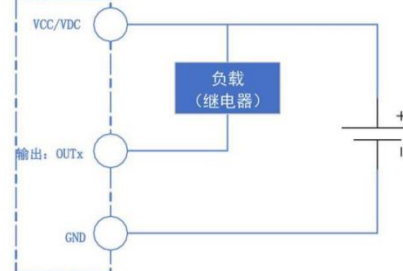
有源传感器:



三线制传感器（有源）:



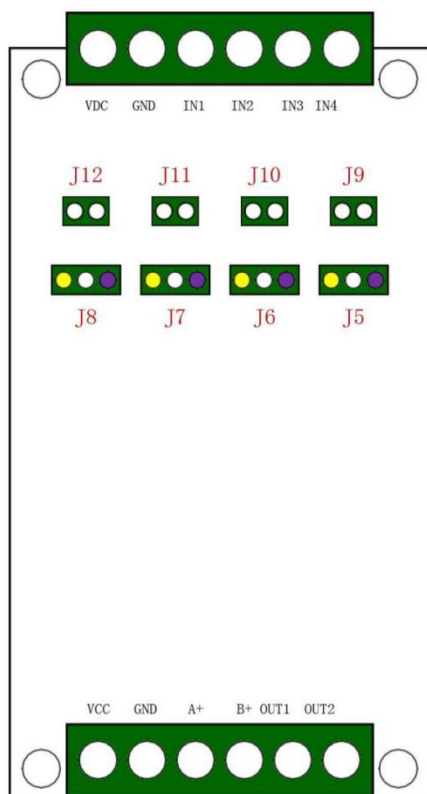
注: 数字输出接继电器示意图:



底部 6 槽接线位:

VDC:	继电器阳极公共端	A+: RS485 通讯线 A	OUT1: 数字输出 DO1
GND:	继电器阴极公共端	B+: RS485 通讯线 B	OUT2: 数字输出 DO2

3.2 跳线



电路板上 有 4 个 跳线座，通过跳线选择不同点量程，结合配置软件实现多量程功能：

·通道 IN1:

0~30V 量程：选择 JP8 紫色侧 2 个跳线座

0~5V 量程：选择 JP8 黄色 2 个跳线座

0~20mA 量程：选择 J12 跳线座

·通道 IN2:

0~30V 量程：选择 JP7 紫色侧 2 个跳线座

0~5V 量程：选择 JP7 黄色 2 个跳线座

0~20mA 量程：选择 P11 跳线座

·通道 IN3:

0~30V 量程：选择 JP6 紫色侧 2 个跳线座

0~5V 量程：选择 JP6 黄色 2 个跳线座

0~20mA 量程：选择 J10 跳线座

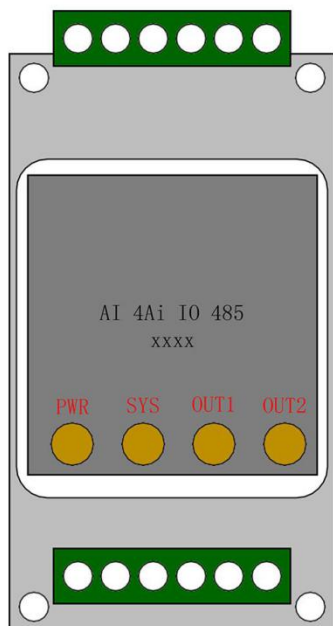
·通道 IN4:

0~30V 量程：选择 JP5 紫色侧 2 个跳线座

0~5V 量程：选择 JP5 黄色 2 个跳线座

0~20mA 量程：选择 J9 跳线座

3.3 LED 指示灯



4 个 LED 指示灯：

·PWR:	正常供电时常亮
·SYS:	系统状态灯，正常运行时每秒闪烁一次
·OUT1:	输出通道 DO1 状态指示灯， 亮：输出信号、灭：无输出
·OUT2:	输出通道 DO2 状态指示灯， 亮：输出信号、灭：无输出

四、软件操作

设备参数配置教程，结合《用户测试文档》即可对设备进行简单测试

4.1 配置软件

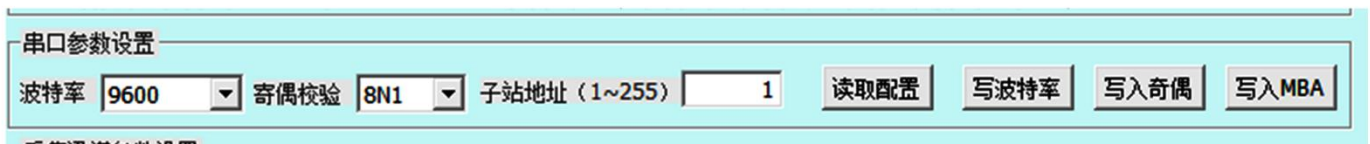
参数配置软件介绍：

参数配置准备：

- (1) 用 USB-485 工具连接设备到电脑
- (2) 在串口配置框内配置串口波特率、停止位、校验位、数据位；（默认波特率 9600，数据位 8，停止位 1，校验位 None），打开串口。（每次设置成功都会弹出对话框，然后点确认，表示连接没有问题，写入数据成功）
- (3) 点击“读配置”，可将默认的配置信息读取进数据框。

4.2配置基本参数

Modbus 地址：Modbus 地址参数



4.3DO 输出相关参数

DO1 初始值：点击“DO通道单独输出设置”，通过选择状态可配置上电后DO1的输出状态

DO2 初始值：点击“DO通道单独输出设置”，通过选择状态可配置上电后DO2的输出状态



4.4AI 模拟量采集相关参数

AI1 采集量程：点击“采样量程”栏，选择对应选项修改 AI 采集量程，写入量程，弹出成功写入的对话框。

AI2 采集量程：点击“采样量程”栏，选择对应选项修改 AI 采集量程，写入量程，弹出成功写入的对话框。

AI3 采集量程：点击“采样量程”栏，选择对应选项修改 AI 采集量程，写入量程，弹出成功写入的对话框。

AI4 采集量程：点击“采样量程”栏，选择对应选项修改 AI 采集量程，写入量程，弹出成功写入的对话框。

4.4 AI 模拟量采集相关参数

采集通道参数设置

AI1通道参数设置

ADC值	2086.35	范围min	0	报警min	5	DO关联	不操作DO	读配置	范围MIN	报警MIN	写入DO
实际值	10.1897	范围max	20	报警max	8	采样量程	电压0~10v	测读值	范围MAX	报警MAX	写入量程

AI2通道参数设置

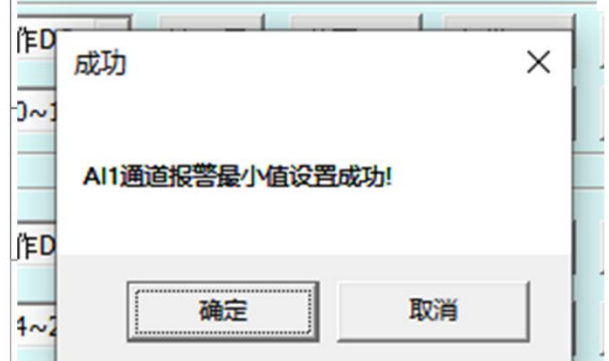
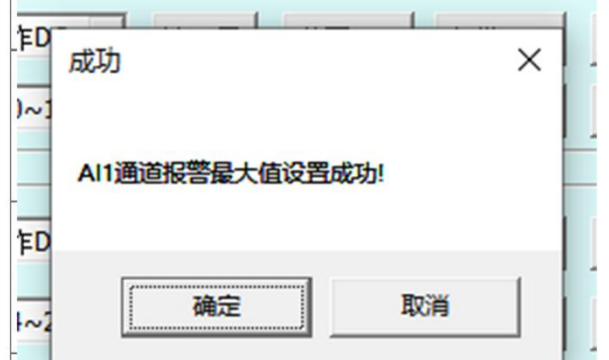
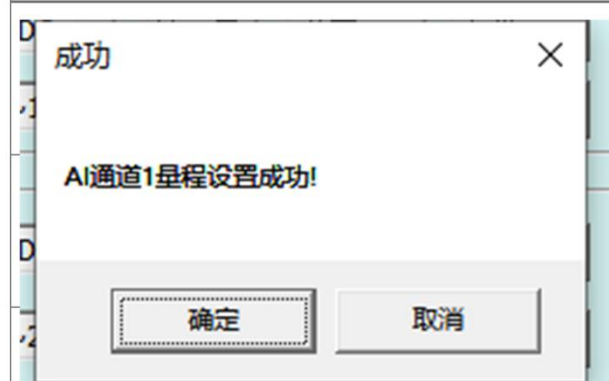
ADC值	2605.72	范围min	0	报警min	5	DO关联	不操作DO	读配置	范围MIN	报警MIN	写入DO
实际值	12.7264	范围max	20	报警max	8	采样量程	电流4~20m	测读值	范围MAX	报警MAX	写入量程

AI3通道参数设置

ADC值	2605.72	范围min	0	报警min	5	DO关联	不操作DO	读配置	范围MIN	报警MIN	写入DO
实际值	12.7264	范围max	20	报警max	8	采样量程	电流4~20m	测读值	范围MAX	报警MAX	写入量程

AI4通道参数设置

ADC值	2605.72	范围min	0	报警min	5	DO关联	不操作DO	读配置	范围MIN	报警MIN	写入DO
实际值	12.7264	范围max	20	报警max	8	采样量程	电流4~20m	测读值	范围MAX	报警MAX	写入量程



AI1 告警操作 DO 端口：在报警栏内输入最小报警值和最大报警值，点击选项框“报警MIN”和选项框“报警MAX”，分别弹出对话框表示参数已经写入成功。

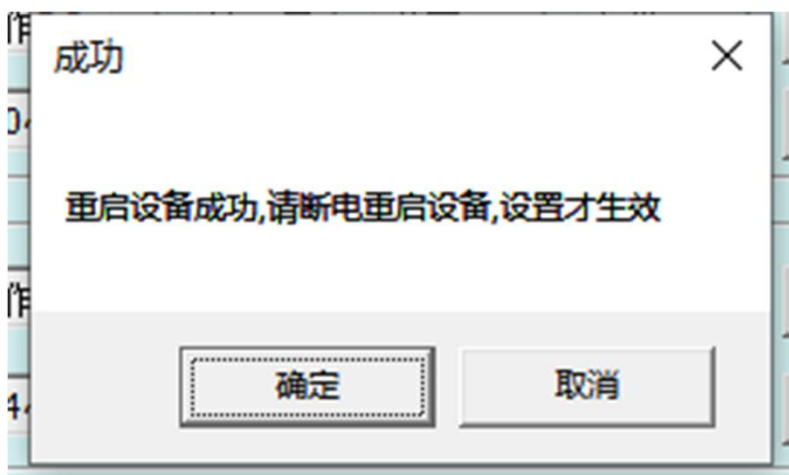
AI2 告警操作 DO 端口：在报警栏内输入最小报警值和最大报警值，点击选项框“报警MIN”和选项框“报警MAX”，分别弹出对话框表示参数已经写入成功。

AI3 告警操作 DO 端口：在报警栏内输入最小报警值和最大报警值，点击选项框“报警MIN”和选项框“报警MAX”，分别弹出对话框表示参数已经写入成功。

AI4 告警操作 DO 端口：在报警栏内输入最小报警值和最大报警值，点击选项框“报警MIN”和选项框“报警MAX”，分别弹出对话框表示参数已经写入成功。

4.5 AI 模拟通道参数校准参数

每次修改参数，都需要点击重启设备，模块设备断电重启，才能生效。



五、MODBUSRTU 通讯协议、组态软件软件说明

5.1 通讯协议

本产品支持标准 Modbus RTU 从站协议，能够支持标准 Modbus RTU 组态软件，详细介绍参《rtu_modbus 协议手册》v1.01 版

5.2 寄存器地址

寄存器地址	名称	字节数	说明	备注
模拟量输入				
0000	AI1_H	2	模拟量通道 1 高	每个模拟量通道占 2 个 Modbus 寄存器，4 个字节，格式为浮点数，浮点数格式符合 IEEE 754 标准
0001	AI1_L	2	模拟量通道 1 低	
0002	AI2_H	2	模拟量通道 2 高	
0003	AI2_L	2	模拟量通道 2 低	
0004	AI3_H	2	模拟量通道 3 高	
0005	AI3_L	2	模拟量通道 3 低	
0006	AI4_H	2	模拟量通道 4 高	
0007	AI4_L	2	模拟量通道 4 低	
数字量输出				
00014	D01	2	数字量通道 1	0000 表示断开 0001 表示闭合
00015	D02	2	数字量通道 2	

5.2 Modbus RTU 功能码

功能码	操作	说明
01	读取单位 DO 状态	Bit 位表示 DO 输出状态
03	读取 AI, DO 寄存器值	读取 AI, DO 寄存器值
04	读取 AI, DO 寄存器值	读取 AI, DO 寄存器值
05	写单个 DO	0xFF00: 闭合;0x0000: 断开
06	写单个 DO	0x0001: 闭合;0x0000: 断开
0F	写多个 DO	参照《rtu_modbu 协议手册》v1.01
10	写多个 DO	参照《rtu_modbu 协议手册》v1.01

详细讲解参照《rtu_modbus 协议手册》v1.01

6.1 功能码描述

6.1.1 01 读线圈

可以使用此功能码读取继电器 DO1~DO2 的状态。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-2 为 0-1。

响应 PDU 中 N 个字节的线圈状态的每一个 bit 位代表一个线圈的状态，状态1=ON, 0=OFF。第一个字节的最低位 LSB 代表第 0 号线圈的状态（即起始地址指定的线圈号为 0 号线圈），其他线圈依次类推，一直到这个字节的最高位 MSB 为止，并且后续字节中都是由低到高代表连续的各线圈状态。如果线圈数量不是 8 的倍数，将用零填充剩余最后数据字节中的剩余比特，字节数量域说明了数据的完整字节数。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x01
起始地址	2 个字节	0x0014 至 0x0015
线圈数量	2 个字节	n(1 至 2)
CRC 校验	2 个字节	

注: 线圈状态的字节数 $N = \text{线圈数量 } n / 8$, 如果余数不等于 0, 则 $N = n / 8 + 1$

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x81 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读离散量 DO1-DO4 的实例

请求		响应	
地址	64	地址	64
功能码	01	功能码	01
起始地址高 H	00	字节数	01
起始地址低 L	14	DO1-DO2 状态	01
线圈数量高 H	00	CRC 校验高 H	C4
线圈数量低 L	02	CRC 校验低 L	34
CRC 校验高 H	F4		
CRC 校验低 L	3A		

发送: 640100140002F43A DTU 响应: 64010101C434

DO1-DO2 的状态字节为 0D, 二进制 00000001, DO1 是这个字节的 LSB(第 0 位)为 1 表示闭合, DO2 是第 1 位为 0 表示断开, 用 0 填充剩余的 6 位。

6.1.2 03 读保持寄存器

6.1.3 04 读输入寄存器

使用该功能码可以读取所有寄存器包括 AI1-AI4、DO1-DO2 的状态。

请求 PDU 详细说明了起始寄存器地址和寄存器数量，从零开始寻址寄存器，因此寻址寄存器 1-16 为 0-15。

响应报文中的寄存器数据每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2。对于 AI1-AI4，一个通道占用 2 个寄存器，4 个字节的值使用浮点数表示，对于 DO1-DO2，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 04
起始地址	2 个字节	0x0000 至 0x0017
寄存器数量	2 个字节	n(1 至 24)
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x03 或 0x04
字节数	1 个字节	$N=2*n$
寄存器值	N 个字节	$N=2*n$, n 为寄存器数量
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x83 或 0x84 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个读模拟量输入 AI1-AI8、离散量输入 DI1-DI4、继电器状态 DO1-DO4 的

请求		响应	
地址	64	地址	64
功能码	03	功能码	03
起始地址高 H	00	字节数	30
起始地址低 L	00	AI1-AI4 值	16 个字节
寄存器数量高 H	00	填充字节	24 个字节
寄存器数量低 L	18	DO1-DO2 状态	4 个字节

CRC 校验高 H	4C	填充字节	4 个字节
CRC 校验低 L	35	CRC 校验高 H	
		CRC 校验低 L	

发送: 6403000000184C35

6.1.4 05 写单个线圈

可以使用该功能码写单个继电器 DO1-DO2 为断开或闭合

请求数据域中的常量说明请求的 ON/OFF 状态, 十六进制值 0xFF00 请求输出为 ON(闭合), 十六进制值 0x0000 请求输出为 OFF(断开), 其他所有值都是非法的, 对输出不起作用, DTU 返回错误响应。

请求域中的输出地址规定了要写入线圈的地址。

正常响应是请求的应答, 在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x0015
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x05
输出地址	2 个字节	0x0014 至 0x0015
输出值	2 个字节	0x0000 或 0xFF00
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x85 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求写线圈 DO2 为 ON(闭合)的实例

请求		响应	
地址	64	地址	64
功能码	05	功能码	05
输出地址高 H	00	输出地址高 H	00

输出地址低 L	15	输出地址低 L	15
输出值高 H	FF	输出值高 H	FF
输出值低 L	00	输出值低 L	00
CRC 校验高 H	94	CRC 校验高 H	94
CRC 校验低 L	0B	CRC 校验低 L	0B

发送：64050015FF00940B

DTU 响应：64050015FF00940B

6.1.5 06 写单个寄存器

可以使用该功能码写单个继电器 DO1-DO2 为断开或闭合。

请求数据域中的寄存器值说明请求的 ON/OFF 状态，十六进制值 0001 请求输出为 ON(闭合)，十六进制值 0x0000 请求输出为 OFF(断开)。

请求域中的寄存器地址规定了要写入线圈的地址。

正常响应是请求的应答，在写入线圈状态后返回这个正常响应。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x0015
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x06
寄存器地址	2 个字节	0x0014 至 0x0015
寄存器值	2 个字节	0x0000 至 0xFFFF
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x86 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求写线圈 DO2 为 ON(闭合)的实例

请求		响应	
地址	64	地址	64
功能码	06	功能码	06

寄存器地址高 H	00	寄存器地址高 H	00
寄存器地址低 L	15	寄存器地址低 L	15
寄存器值高 H	00	寄存器值高 H	00
寄存器值低 L	01	寄存器值低 L	01
CRC 校验高 H	50	CRC 校验高 H	50
CRC 校验低 L	3B	CRC 校验低 L	3B

发送: 640600150001503B

DTU 响应: 640600150001503B

6.1.6 0F 写多个线圈

可以使用此功能码写多个继电器 DO1~DO4 为断开或闭合。

请求 PDU 详细说明了起始地址，即指定第一个线圈的地址和线圈数量，从零开始寻址线圈，因此寻址线圈 1-2 为 0-1。

请求数据域中的内容说明了被请求的 ON/OFF 状态，域比特位中的逻辑“1”请求相应输出为 ON，域比特位中的逻辑“0”请求相应输出为 OFF。从数据域中第一个字节的 bit0 开始到 bit7，然后到第二个字节的 bit0，依次表示第一个线圈到第 n 个线圈的 ON/OFF 值。

正常响应返回功能码、起始地址和线圈数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x0015
线圈数量	2 个字节	n(1 至 2)
字节数	1 个字节	$N=n/8$, 或 $N=n/8+1$
输出值	N 个字节	
CRC 校验	2 个字节	

注: 线圈输出字节数 $N=线圈数量 n/8$, 如果余数不等于 0, 则 $N=n/8+1$

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x0F
起始地址	2 个字节	0x0014 至 0x0017
线圈数量	2 个字节	n(1 至 2)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x8F (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个请求从线圈 DO1 开始写入 4 个线圈的实例

请求		响应	
地址	64	地址	64
功能码	0F	功能码	0F
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
线圈数量高 H	00	线圈数量高 H	00
线圈数量低 L	02	线圈数量低 L	02
字节数	01	CRC 校验高 H	9D
输出值	02	CRC 校验低 L	FB
CRC 校验高 H	68		
CRC 校验低 L	82		

发送：640F001400020103A942 DTU 响应：640F001400029DFB

DO1-DO2 的输出值为 02，二进制 00000010，DO1 是这个字节的 LSB(第 0 位) 为 0 表示断开，DO2 是第 1 位为 1 表示闭合，用 0 填充剩余未使用的 6 位。

6.1.7 10 写多个寄存器

使用该功能码可以写连续寄存器 DO1-DO2 的状态。

请求 PDU 详细说明了起始寄存器地址、寄存器数量、字节数和寄存器值，从零开始寻址寄存器，因此寻址寄存器 1-16 为 0-15。

寄存器数据中每个寄存器有 2 个字节，对于每一个寄存器，第一个字节代表寄存器值的高位，第二个字节代表寄存器值的低位。字节数为寄存器数量乘以 2，2 个字节的值 0000 代表继电器断开，0001 代表继电器闭合。

正常响应返回功能码、起始地址和被写入寄存器的数量。

请求 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x0015
寄存器数量	2 个字节	n(1 至 2)
字节数	1 个字节	$N=2*n$
寄存器值	N 个字节	$N=2*n$, n 为寄存器数量
CRC 校验	2 个字节	

响应 PDU

地址	1 个字节	
功能码	1 个字节	0x10
起始地址	2 个字节	0x0014 至 0x0015
寄存器数量	2 个字节	n(1 至 2)
CRC 校验	2 个字节	

错误响应 PDU

地址	1 个字节	
功能码	1 个字节	0x90 (请求功能码+0x80)
异常码	1 个字节	0x01 或 0x02 或 0x03 或 0x04
CRC 校验	2 个字节	

这是一个控制继电器 DO1-DO4 的实例

请求		响应	
地址	64	地址	64
功能码	10	功能码	10
起始地址高 H	00	起始地址高 H	00
起始地址低 L	14	起始地址低 L	14
寄存器数量高 H	00	寄存器数量高 H	00
寄存器数量低 L	02	寄存器数量低 L	02
字节数	04	CRC 校验高 H	08
DO1 寄存器值高 H	00	CRC 校验低 L	39
DO1 寄存器值高 L	01		
DO2 寄存器值高 H	00		
DO2 寄存器值高 L	00		
DO3 寄存器值高 H	4D		
DO3 寄存器值高 L	5D		
DO4 寄存器值高 H	64		
DO4 寄存器值高 L	10		
CRC 校验高 H	00		
CRC 校验低 L	14		

发送: 64100014000204000100004D5D

DTU 响应: 6410001400020839

DO1 寄存器值为 0001 表示闭合, DO2 寄存器值为 0000 表示断开

6.2 错误码描述

错误码含义：当 DTU 收到错误的 Modbus 指令时，会返回功能码为请求功能码+0x80，紧随着一个字节的错误码代表出错原因。

错误码 01：表示不支持的功能码，众山 DTU 支持上述 8 种功能码，除此之外的功能码都会返回错误码为 01 的错误。

错误码 02：表示起始地址不存在或者起始地址加上寄存器数量后的地址不存在。总的来说表示访问的寄存器不存在。

错误码 03：表示寄存器数量不符合规范或者寄存器值非法。

错误码 04：表示读写寄存器错误。

6.3 CRC 校验算法

CRC 即循环冗余校验码（Cyclic Redundancy Check）：是数据通信领域中最常用的一种查错校验码，其特征是信息字段和校验字段的长度可以任意选定。循环冗余检查（CRC）是一种数据传输检错功能，对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

附 CRC 校验算法代码：

```
uint16_t mb_crc( uint8_t *snd, uint16_t num )
{
    uint8_t CRC_Lb, CRC_Hb;
    uint8_t CRC_L, CRC_H;
    uint16_t crc;

    CRC_H = 0xFF;
    CRC_L = 0xFF;

    for ( uint16_t i = 0; i < num; i++ )
    { CRC_L = CRC_L ^ snd[i];
      for ( uint16_t j = 0; j < 8; j++ )
      { CRC_Lb = CRC_L;
        if ((CRC_L & 1) == 1)
        { CRC_L = (CRC_L - 1)
          / 2; CRC_Lb = CRC_L;
          CRC_Hb = CRC_H;
          if ( (CRC_H & 1) == 1 )
          { CRC_L = CRC_L + 128;
            CRC_Lb = CRC_L;
            CRC_H = (CRC_H - 1) / 2;
            CRC_Hb = CRC_H;
          } else {
            CRC_H = CRC_H / 2;
            CRC_Hb = CRC_H;
          }
        }
      }
    }
}
```

```
    CRC_L = CRC_L / 2;
    CRC_Lb = CRC_L;
    CRC_H = CRC_H ^ 0xA0;
    CRC_Hb = CRC_H;
}
else
{
    CRC_L = CRC_L / 2;
    CRC_Lb = CRC_L;
    CRC_Hb = CRC_H;
    if ( (CRC_H & 1) == 1 )
    { CRC_L = CRC_L + 128;
      CRC_Lb = CRC_L;
      CRC_H = (CRC_H - 1) / 2;
      CRC_Hb = CRC_H;
    } else {
      CRC_H = CRC_H / 2;
      CRC_Hb = CRC_H;
    }
}
}

crc = CRC_L;
crc <<= 8;
crc |= CRC_H;
return crc;
}
```