

## 1. WM-IO44功能简介

WM-IO44是带隔离数字量输入，继电器输出模块。模块有4路继电器输出其中2路B型（常开），2路A型（常闭）。另外4路隔离数字量输入，支持开关触点信号和电平信号。支持标准的Modbus协议, 适用于采集工业现场的数字量信号以及控制小功率电气设备, 模块的外观如图 1.1所示。



图 1.1 WM-IO44 外观 端子排列示意图

## 1.1 主要技术指标

### 1.1.1 数字量输入

- ◆ 输入路数：4路（光电隔离）
- ◆ 输入类型：开关触点信号或电平信号
- ◆ 隔离电压：2500V<sub>DC</sub>
- ◆ 触点输入阻抗：≤2kΩ
- ◆ 电平输入范围：
- ◆ 高电平（数字1）： +4.0 V~+30V
- ◆ 低电平（数字0）： ≤+1V

### 1.1.2 继电器输出

- ◆ 输出路数： 4路
- ◆ 输出类型： 2路B型（默认是 常开），2路A型（常闭）
- ◆ 输出负载： 110V DC 1A  
125V AC 1A
- ◆ 接触电阻： 100mΩ
- ◆ 绝缘强度：触点与线圈之间 2000V<sub>rms</sub>（1分钟）

- ◆ 触点与触点之间 1000Vrms (1分钟)
- ◆ 机械寿命: 100 000 000次
- ◆ 电气寿命: 100 000次 @30VDC 2A  
500 000次 @30VDC 1A
- ◆ 最大操作时间: 5ms
- ◆ 最大释放时间: 5ms

### 1.1.3 系统参数

- ◆ 供电电压: +9~+30V<sub>DC</sub>, 电源反接保护
- ◆ 工作温度范围: -20℃~+85℃
- ◆ 塑料外壳, 标准DIN导轨安装
- ◆ 通讯接口: 隔离 2500 V<sub>DC</sub>, ESD、过压、过流保护

## 1.3 端子信息

WM-IO44 模块的外观如图 1.1 所示 共有 20 个端子, 壳体上端子排列如图 1.1 所示。

### 1.3.2 端子描述

WM-IO44 的端子定义说明如下:

- ◆ GND, VCC 为模块的电源输入端, GND 接电源负端, VCC 接电源正端;
- ◆ INIT 为模块的默认通信参数硬件使能端子, 当此端子接地, 模块将以默认的通信参数进行初始化, 并且通信参数可配置;
- ◆ 485GND, 485+, 485-为隔离的 RS-485 接口端子, 485GND 为接口的隔离地, 485+接 RS-485 收发器的 A 端, 485-接 RS-485 收发器的 B 端。
- ◆ DI0~DI3 为模块的 4 路数字量输入通道端子。
- ◆ D00~D03 为模块的 4 路继电器输出通道端子。
- ◆ IN-GND 为数字量输入端口的公共地, 与模块的 DI0~DI3 为同一个地, 它们和 485GND 之间是电气隔离的, 隔离电压可达 2500 VDC 。

## 1.1 信号指示灯

PWR 为电源指示灯 (红色) 和工作状态指示灯 RUN (色) 485 通信 (TXD) 发送指示灯, (RXD) 接收指示灯, (D00~D03) 4 路继电器输出通道指示灯。

## 1.2 电源和通讯线的连接

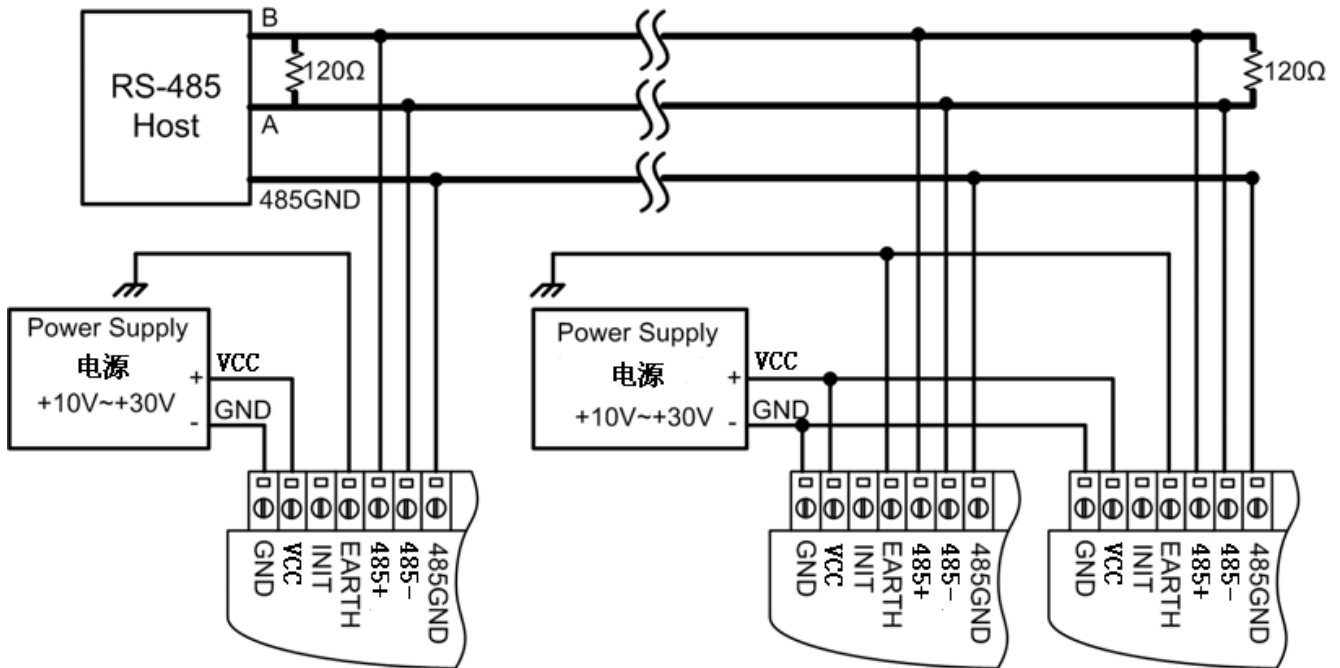
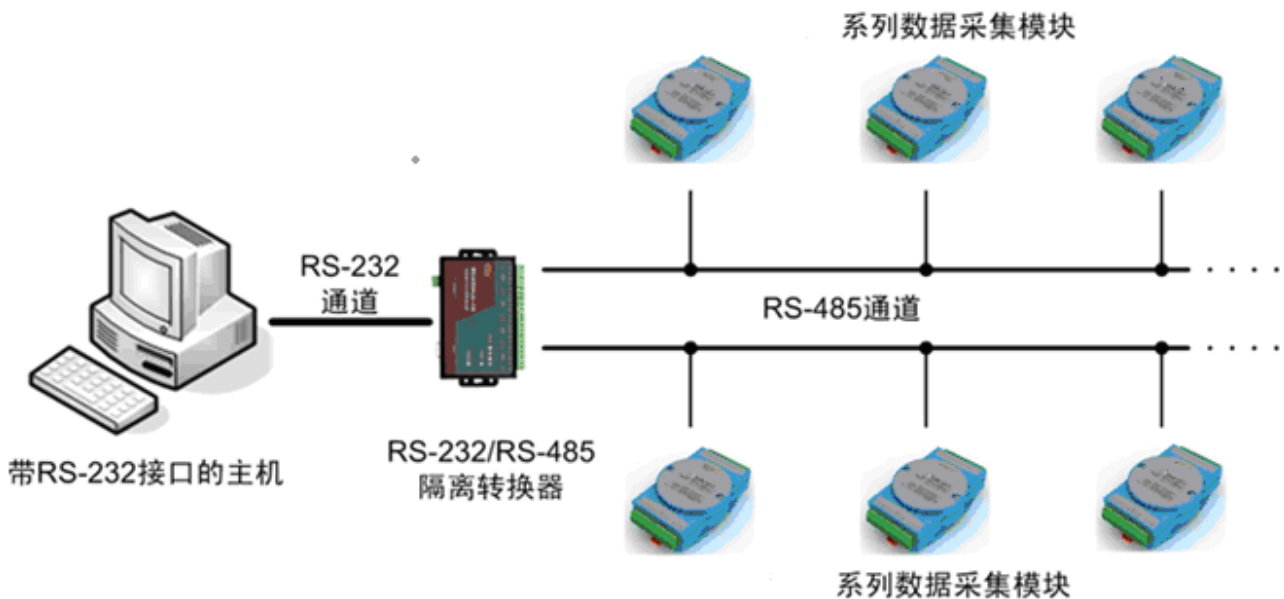


图 1.2 电源和通讯线的连接



## WM-IO44 的输入输出功能

在工业控制过程中，经常需要对现场的数字量输入信号进行采集，并通过输出数字量控制现场的执行器或开关设备。

**WM-IO44** 模块具有4路数字量输入通道和4路继电器输出通道，可以广泛的应用到各种工业领域。模块输出数据具有安全保护功能，可以在模块和主机失去通讯的时候，输出用户设定的一组输出状态，以保证受控设备的安全性。

### 2.1 数字量输入

**WM-IO44** 的4路隔离数字量输入通道，可以用来检测开关触点信号和电平信号，输入信号逻辑状态定义

如表 2.1所示。

表 2.1 输入信号定义输入信号类型		信号定义
电压型数字量输入信号	高电平信号	状态1, 电压范围: +4.0V~+30V
	低电平信号	状态0, 电压范围: ≤+1V
开关触点型数字量输入信号	开路触点信号	状态1
	闭合触点信号	状态0

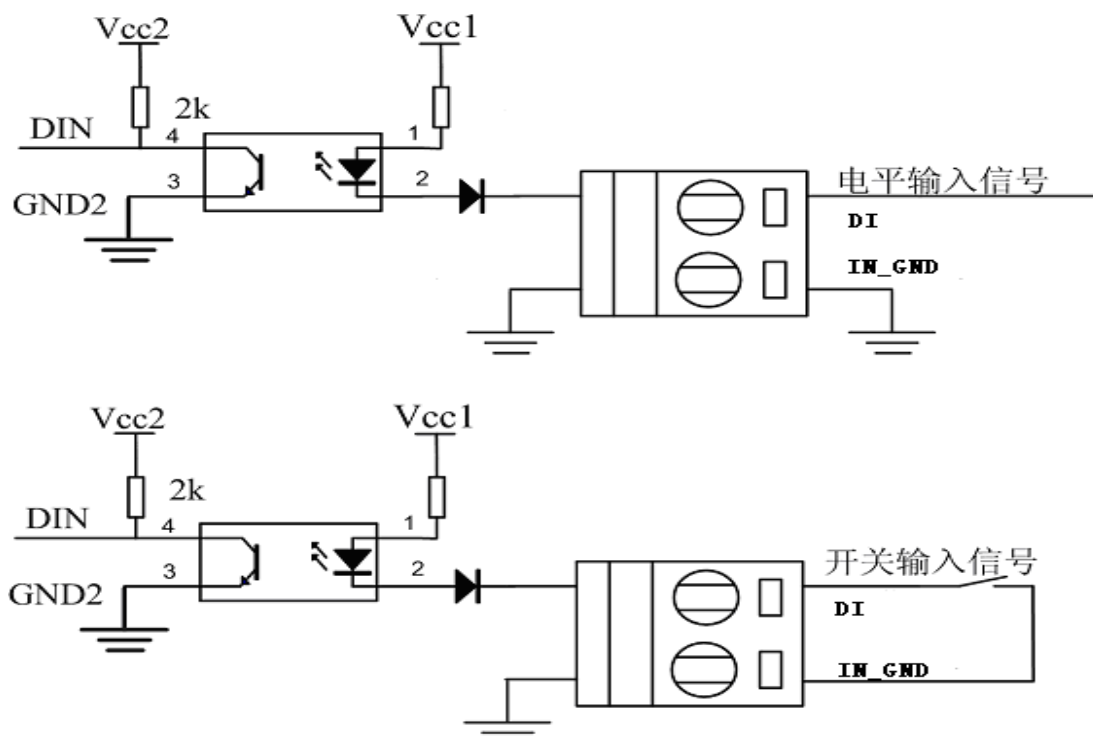


图 2.1 数字量输入接线方式

当以电平作为数字量输入时,  $V_{in} = L (<1.0V)$  隔离光耦导通, 光耦输出端  $DIN = L$ , 当输入  $V_{in} = H (+4.0 \sim +30V)$  时, 光耦不导通, 光耦输出  $DIN = H$ 。

**WM-IO44** 具有 4 路继电器输出, 能控制外部 4 个开关设备, 工业上用 WM-IO44 的常开触点输出通道驱动固态继电器的接线如上图 2.4 所示。用户可以用 DOA 端和 DOB 端构成单开关, 可以根据实际情况接线。

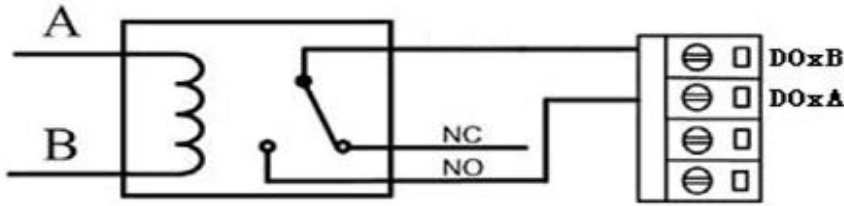


图 2.3 (b) 型继电器输出 (n=3, 4)

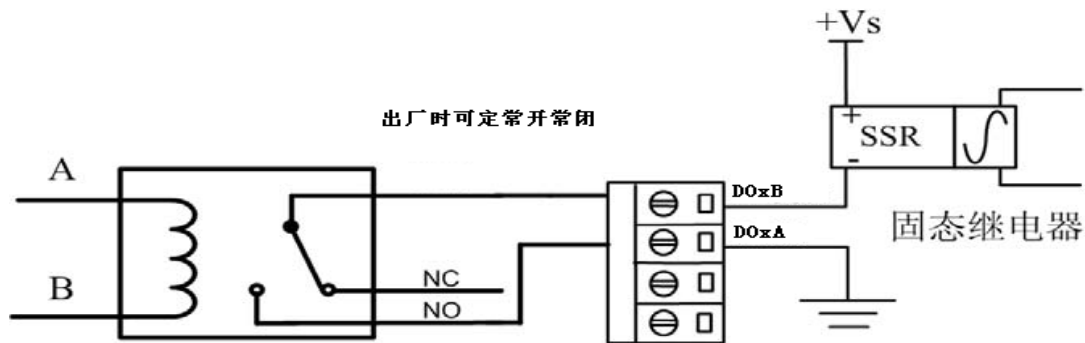


图 2.4 加固态继电器接线方式

### 3.WM-IO44资源地址及通信协议

#### 3.1 WM-IO系列模块资源地址

WM-IO系列模块的资源地址分配如表 4.1所示：

表 4.1 模块资源地址分配表资源地址			功能说明
I/O资源	0x0000-0x0007	DI输入地址单元0000-0003 DO输出地址单元0003-0007	离散量线圈(位)地址单元的数据长度为1个字节(8bit)
	0x000A	DI/O输入输出单元	
配置资源	0x0000-0x0009	I/O配置资源	每个地址单元的数据长度为2个字节(16bit)

配置资源用于存放模块的配置参数，不同功能模块的配置资源不同，其编号范围为 0x0000~0x00ff。

#### 1. 资源地址

WM-IO44模块I/O资源寄存器地址（0x000A）地址单元的数据长度为1个字节（8bit），具有DI 4路数字量输入通道，其DO 4路继电器输出。

（0x000A）为资源寄存器地址可用 **Modbus-RTU** 功能号（03, 10）直接读写，当采用离散量线圈（位）进行读写操作时可用 **Modbus-RTU** 功能号（01, 05）离散量线圈(位)地址：0000--0007，其地址如表 4.5 所示。

表 4.5 离散量线圈地址

离散量线圈(位)地址	0007	0006	0005	0004	0003	0002	0001	0000
------------	------	------	------	------	------	------	------	------

寄存器地址0x000A	DI3	DI2	DI1	DI0	DO3	DO2	DO1	DO0
-------------	-----	-----	-----	-----	-----	-----	-----	-----

WM-IO44模块Modbus-RTU协议功能号（10）写寄存器（注意：禁止对寄存器地址为（0x000A）字节写操作），因为 DO和DI映射到一个寄存器地址为（0x000A）地址单元的数据长度为1个字节（8bit）。

### 3.2 通信参数设置

WM-IO系列模块支持标准的Modbus-RTU协议，具有多种通信传输模式可供配置。模块的通信参数如：设备地址、波特率、数据位长度和奇偶校验方式等也可通过配置软件进行配置。通信参数都是保存在模块的E2PROM中，用户可以通过RS-485接口进行远程软件配置。

要通过配置软件进行修改通信参数，用户首先需要知道该模块的参数配置。由于模块没有诸如拨码开关之类的硬件设置来指示此时的参数配置，可能会存在用户忘了某个WM-IO模块的通信参数的情况。为了解决此问题，每个WM-IO模块都有一个硬件使能输入端子INIT。将此端子连接到GND后，给模块上电，模块的通信参数处于确定的(默)状态：

地址：1

波特率：9600bps

通信协议传输模式：Modbus-RTU

数据格式：无奇偶校验，8个数据位，1个停止位

将INIT端子与GND短接，模块用以上确定的通信参数进行初始化，并不会改变E2PROM中保存的配置参数。但只有在这个条件下，通信配置参数才可以进行修改，否则对通信参数的配置命令都将回应异常响应。

通信参数修改后，必须把INIT端与GND断开连接后，给模块重上电或通过软件复位模块，配置的通信参数才生效。

### 3.3 通信协议

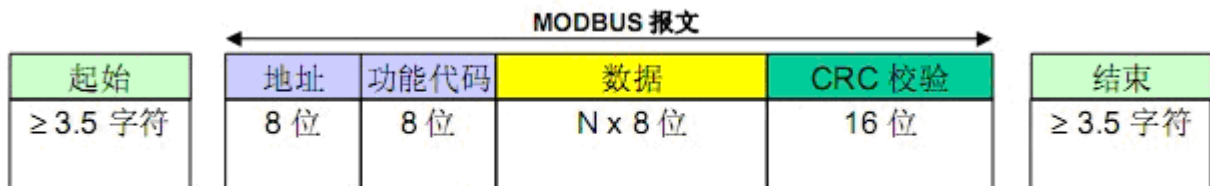


图 2.5 RTU 报文帧

WM-IO44模块支持标准的Modbus-RTU协议的功能号：（01）读位，（05）写位，（03）读寄存器，（10）写寄存器

功能号（01）读位：

01 01 00 00 00 01 读位DO0	01 01 00 04 00 01 读位DI0
01 01 00 01 00 01 读位DO1	01 01 00 05 00 01 读位DI1
01 01 00 02 00 01 读位DO2	01 01 00 06 00 01 读位DI2
01 01 00 03 00 01 读位DO3	01 01 00 07 00 01 读位DI3

功能号（05）写位：

01 05 00 00 00 00 写位DO0	01 05 00 00 FF 00 写位DO0
01 05 00 01 00 00 写位DO1	01 05 00 01 FF 00 写位DO1
01 05 00 02 00 00 写位DO2	01 05 00 02 FF 00 写位DO2
01 05 00 03 00 00 写位DO3	01 05 00 03 FF 00 写位DO3



功能号 (03) 读寄存器:

01 03 00 0A 00 01 读寄存器

功能号 (10) 写寄存器:

01 10 00 00 00 0A 01 00 00 **注意:** 写寄存器 (禁止对地址为 (0x000A) 寄存器字节的写操作禁用)

## 功能码描述

### 6.1 01 (0x01) 读线圈

在一个远程设备中, 使用该功能码读取线圈的 1 至 2000 连续状态。请求 PDU 详细说明了起始地址, 即指定的第一个线圈地址和线圈编号。从零开始寻址线圈。因此寻址线圈 1-16 为 0-15。

根据数据域的每个比特将响应报文中的线圈分成为一个线圈。指示状态为 1= ON 和 0= OFF。第一个数据字节的 LSB (最低有效位) 包括在询问中寻址的输出。其它线圈依次类推, 一直到这个字节的高位端为止, 并在后续字节中从低位到高位顺序。

如果返回的输出数量不是八的倍数, 将用零填充最后数据字节中的剩余比特 (一直到字节的高位端)。字节数量域说明了数据的完整字节数。

#### 请求 PDU

功能码	1 个字节	0x01
起始地址	2 个字节	0x0000 至 0xFFFF
线圈数量	2 个字节	1 至 2000 (0x7D0)

#### 响应 PDU

功能码	1 个字节	0x01
字节数	1 个字节	N*
线圈状态	N 个字节	n=N 或 N+1

### 6.5 05 (0x05) 写单个线圈

在一个远程设备上，使用该功能码写单个输出为 ON 或 OFF。

请求数据域中的常量说明请求的 ON/OFF 状态。十六进制值 FF 00 请求输出为 ON。十六进制值 00 00 请求输出为 OFF。其它所有值均是非合法的，并且对输出不起作用。

请求 PDU 说明了强制的线圈地址。从零开始寻址线圈。因此，寻址线圈 1 为 0。线圈值域的常量说明请求的 ON/OFF 状态。十六进制值 0XFF00 请求线圈为 ON。十六进制值 0X0000 请求线圈为 OFF。其它所有值均为非合法的，并且对线圈不起作用。

正常响应是请求的应答，在写入线圈状态之后返回这个正常响应。

#### 请求

功能码	1 个字节	0x05
输出地址	2 个字节	0x0000 至 0xFFFF
输出值	2 个字节	0x0000 至 0x00

#### 响应

功能码	1 个字节	0x05
输出地址	2 个字节	0x0000 至 0xFFFF
输出值	2 个字节	0x0000 至 0xFF00

#### 错误

差错码	1 个字节	0x85
异常码	1 个字节	01 或 02 或 03 或 04

这是一个请求写线圈 173 为 ON 的实例：

请求		响应	
域名	(十六进制)	域名	(十六进制)
功能	05	功能	05
输出地址 Hi	00	输出地址 Hi	00
输出地址 Lo	AC	输出地址 Lo	AC
输出值 Hi	FF	输出值 Hi	FF
输出值 Lo	00	输出值 Lo	00



\*N=输出数量/8, 如果余数不等于 0, 那么N = N+1

**错误**

功能码	1 个字节	功能码+0x80
异常码	1 个字节	01 或 02 或 03 或 04

这是一个请求读离散量输出 20-38 的实例:

请求		响应	
域名	(十六进制)	域名	(十六进制)
功能	01	功能	01
起始地址 Hi	00	字节数	03
起始地址 Lo	13	输出状态 27-20	CD
输出数量 Hi	00	输出状态 35-28	6B
输出数量 Lo	13	输出状态 38-36	05

将输出 27-20 的状态表示为十六进制字节值 CD, 或二进制 1100 1101。输出 27 是这个字节的 MSB, 输出 20 是 LSB。

通常, 将一个字节内的比特表示为 MSB 位于左侧, LSB 位于右侧。第一字节的输出从左至右为 27 至 20。下一个字节的输出从左到右为 35 至 28。当串行发射比特时, 从 LSB 向 MSB 传输: 20... 27、28... 35 等等。

在最后的的数据字节中, 将输出状态 38-36 表示为十六进制字节值 05, 或二进制 0000 0101。输出 38 是左侧第六个比特位置, 输出 36 是这个字节的 LSB。用零填充五个剩余高位比特。

 **注:** 用零填充五个剩余比特 (一直到高位端)。

**6.3 03 (0x03) 读保持寄存器**

在一个远程设备中, 使用该功能码读取保持寄存器连续块的内容。请求 PDU 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。

将响应报文中的寄存器数据分成每个寄存器有两字节, 在每个字节中直接地调整二进制内容。

对于每个寄存器, 第一个字节包括高位比特, 并且第二个字节包括低位比特。

**请求**

功能码	1 个字节	0x03
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 125 (0x7D)

**响应**

功能码	1 个字节	0x03
字节数	1 个字节	2×N*
寄存器值	N*×2 个字节	

\*N = 寄存器的数量

**错误**

差错码	1 个字节	0x83
异常码	1 个字节	01 或 02 或 03 或 04

这是一个请求读寄存器 108-110 的实例：

请求		响应	
域名	(十六进制)	域名	(十六进制)
功能	03	功能	03
高起始地址	00	字节数	06
低起始地址	6B	寄存器值 Hi (108)	02
高寄存器编号	00	寄存器值 Lo (108)	2B
低寄存器编号	03	寄存器值 Hi (109)	00
		寄存器值 Lo (109)	00
		寄存器值 Hi (110)	00
		寄存器值 Lo (110)	00
			64

将寄存器 108 的内容表示为两个十六进制字节值 02 2B，或十进制 555。将寄存器 109-110 的内容分别表示为十六进制 00 00 和 00 64，或十进制 0 和 100。

**6.8 16 (0x10) 写多个寄存器**

在一个远程设备中，使用该功能码写连续寄存器块(1 至约 120 个寄存器)。在请求数据域中说明了请求写入的值。每个寄存器将数据分成两字节。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

功能码	1 个字节	0x10
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	0x0001 至 0x0078
字节数	1 个字节	2×N*
寄存器值	N*×2 个字节	值

\*N=寄存器数量

响应 PDU

功能码	1 个字节	0x10
起始地址	2 个字节	0x0000 至 0xFFFF
寄存器数量	2 个字节	1 至 123 (0x7B)

错误

差错码	1 个字节	0x90
异常码	1 个字节	01 或 02 或 03 或 04

这是一个请求将十六进制 00 0A 和 01 02 写入以 2 开始的两个寄存器的实例：

请求		响应	
域名	(十六进制)	域名	(十六进制)
功能	10	功能	10
起始地址 Hi	00	起始地址 Hi	00
起始地址 Lo	01	起始地址 Lo	01
寄存器数量 Hi	00	寄存器数量 Hi	00
寄存器数量 Lo	02	寄存器数量 Lo	02
字节数	04		
寄存器值 Hi	00		
寄存器值 Lo	00		
寄存器值 Hi	0A		
寄存器值 Lo	01		
	02		

1. 免责声明

版权

产品用户手册

©2012 ShenZhen Wm science and technology development CO., LTD.

本手册所陈述的产品文本及相关软件版权均属深圳威铭科技开发有限公司所有，其产权受国家法律绝对保护，未经本公司授权，其它公司、单位、代理商及个人不得非法使用和拷贝，否则将受到国家法律的严厉制裁。

#### **修改文档的权利**

深圳威铭科技开发有限公司保留任何时候在不事先声明的情况下对本手册的修改的权力。