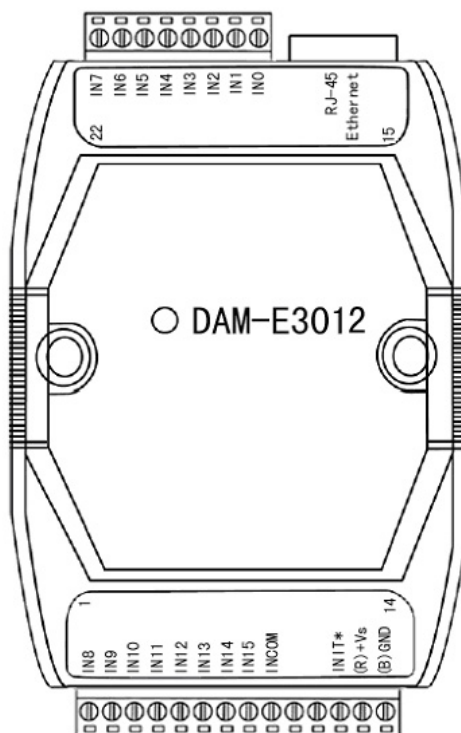


## DAM-E3012 说明书

## ★端子分布图



## ★主要指标

16 路隔离数字量输入模块

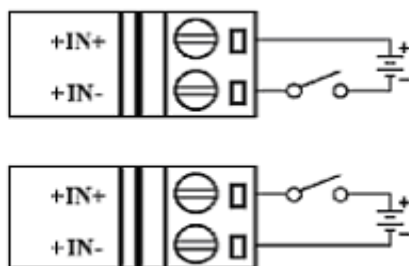
- 通道： 16
- I/O 类型： 16DI
- 输入电平：（湿接点）
  - 逻辑电平 0： +1V（最大）
  - 逻辑电平 1： +4V~+30V
- 隔离电压： 3750V
- 输入通道可作为 500Hz 计数器使用
- 支持 10/100Mbps 以太网
- 提供默认网页或可定制网页
- 内置看门狗

■ 电源：未调理 +10~+30 VDC

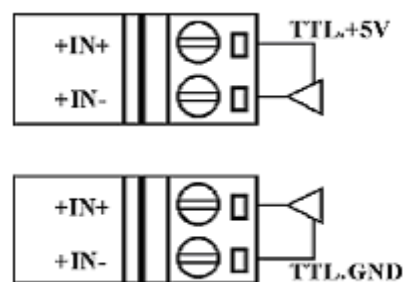
■ 功耗：2W@24V<sub>DC</sub>

★接线图（图的右侧为用户接线方式）

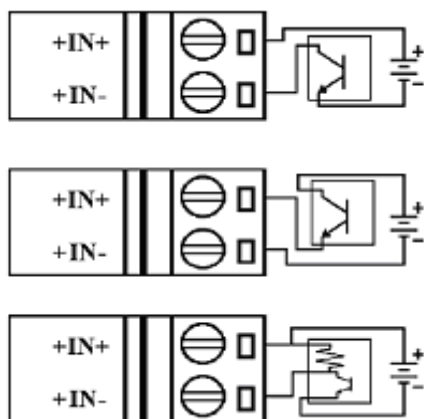
湿接点信号输入



TTL/CMOS 信号输入



集电极开路信号输入



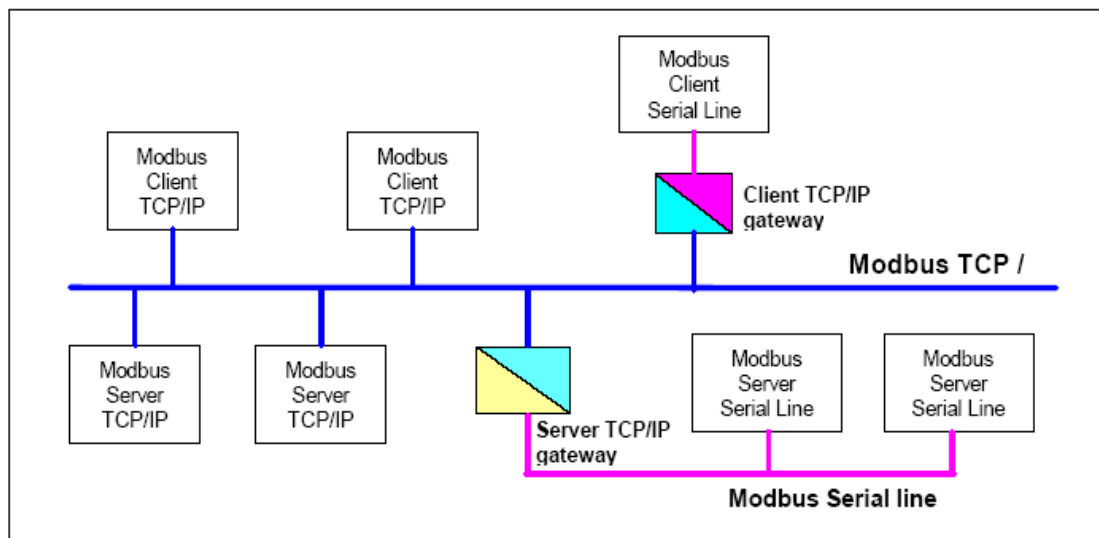
# 通信协议

## 1 MODBUS 简介

工业控制已从单机控制走向集中监控、集散控制，如今已进入网络时代，工业控制器连网也为网络管理提供了方便。MODBUS 就是工业控制器的网络协议中的一种。

MODBUS 规约作为一个通讯协议是由 MODICON 公司在 1979 年首次提出的，它是一个标准的、真正开放的、在工业自动化领域应用最广泛的网络通讯协议。通过此协议，控制器相互之间、控制器经由网络（例如以太网）和其它设备之间可以通信。它已经成为一通用工业标准。通过这一协议，不同厂商生产的控制设备可以连成工业网络，进行集中监控。

Modbus TCP 协议是 MODBUS 协议另一版本，它于 1999 年被开发出来以允许 Internet 用户访问以太网设备。由于没有任何商业利益驱使，Modbus TCP 协议的开放性及其用户对它的熟悉程度再加上其应用的简单易学，现在 Modbus TCP 已经成为世界领先的工业以太网协议。



此协议定义了一个控制器能认识使用的消息结构,而不管它们是经过何种网络进行通信的。它描述了控制器请求访问其它设备的过程,如果回应来自其它设备的请求,以及怎样侦测错误并记录。它制定了消息域格局和内容的公共格式。

当在 Modbus 网络上通信时,此协议决定了每个控制器须要知道它们的设备地址,识别按地址发来的消息,决定要产生何种行动。如果需要回应,控制器将生成反馈信息并用 Modbus 协议发出。在其它网络上,包含了 Modbus 协议的消息转换为在此网络上使用的帧或包结构。这种转换也扩展了根据具体的网络解决节地址、路由路径及错误检测的方法。

标准的 Modbus 口是使用 RS-232C 兼容串行接口,它定义了连接口的针脚、电缆、信号

位、传输波特率、奇偶校验。控制器能直接或经由 Modem 组网。

控制器通信使用主—从技术，即仅设备（**主设备**）能初始化传输（查询）。其它设备（**从设备**）根据**主设备**查询提供的数据做出相应反应。典型的**主设备**：主机和可编程仪表。典型的**从设备**：可编程控制器。

**主设备**可单独和**从设备**通信，也能以广播方式和所有**从设备**通信。如果单独通信，**从设备**返回消息作为回应，如果是以广播方式查询的，则不作任何回应。Modbus 协议建立了**主设备**查询的格式：设备（或广播）地址、功能代码、所有要发送的数据、错误检测域。

**从设备**回应消息也由 Modbus 协议构成，包括确认要行动的域、任何要返回的数据、和错误检测域。如果在消息接收过程中发生错误，或**从设备**不能执行其命令，**从设备**将建立错误消息并把它作为回应发送出去。

在其它网络上，控制器使用对等技术通信，故任何控制都能初始和其它控制器的通信。这样在单独的通信过程中，控制器既可作为**主设备**也可作为**从设备**。提供的多个内部通道可允许同时发生的传输进程。

在消息位，Modbus 协议仍提供了主—从原则，尽管网络通信方法是“对等”。如果控制器发送消息，它只是作为**主设备**，并期望从**从设备**得到回应。同样，当控制器接收到消息，它将建立一**从设备**回应格式并返回给发送的控制器。

### 主设备查询

查询消息中的功能代码告之被选中的**从设备**要执行何种功能。数据段包含了**从设备**要执行功能的任何附加信息。例如功能代码 03 是要求**从设备**读保持寄存器并返回它们的内容。数据段必须包含要告之**从设备**的信息：从何寄存器开始读及要读的寄存器数量。错误检测域为**从设备**提供了一种验证消息内容是否正确的方法。

### 从设备回应

如果**从设备**产生正常的回应，在回应消息中的功能代码是在查询消息中的功能代码的回应。数据段包括了**从设备**收集的数据：像寄存器值或状态。如果有错误发生，功能代码将被修改以用于指出回应消息是错误的，同时数据段包含了描述此错误信息的代码。错误检测域允许**主设备**确认消息内容是否可用。

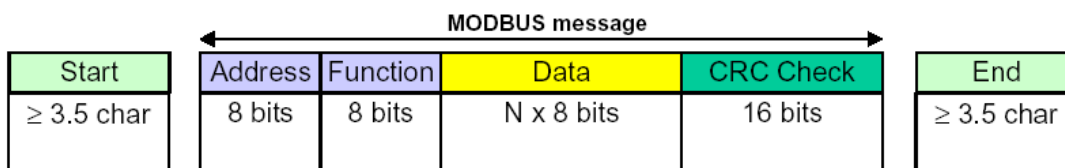
每个 MODBUS 帧都包括地址域 功能域 数据域 错误检测域

## 2 工作方式

## 2.1 MODBUS RTU 方式

地 址	功 能 代 码	数 据 数 量	数 据 1	...	数 据 n	CRC 高字 节	CRC 低字 节
地址域		数据域		错误检测域			

**帧定界:** 在 MODBUS RTU 方式下, 每两个字符之间发送或者接收的时间间隔不能超过 1.5 倍字符传输时间。如果两个字符时间间隔超过了 3.5 倍字符传输时间, 就认为一帧数据已经接收完成, 新的一帧数据传输开始。



## 2.2 MODBUS ASCII 方式

:	地 址	功 能 代 码	数 据 数 量	数 据 1	...	数 据 n	LRC 高 字节	LRC 低 字节	回 车	换 行
地址域		数据域		错误检测域						

**帧定界:** 在 MODBUS ASCII 方式下, 一个 8 位的数据使用两个 ASCII 字符来表示。比如 16 进制的 0x3A 用字符“3”和字符“A”表示。其中“:”表示帧的起始, “CR LF”表示帧的结束。

Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

## 2.3 MODBUS TCP 方式

MBAP Header	功能代 码	数据数 量	数据 1	...	数据 n
协议头	功能域	数据域			

**帧定界:** 在 MODBUS TCP 方式下, 由于模块的地址由 IP 地址确定, 所以不再有地址

域内容，考虑到 TCP 网络是可靠的数据传输网络，故不再有校验数据。但是考虑到在 IP 网上数据到达的顺序可能与我们预期的数据不一致，故增加了一个数据序号，考虑到在 MODBUS TCP 协议上承载 MODBUS 协议，还在头部数据中增加了一个地址域。

Fields	Length	Description -	Client	Server
Transaction Identifier	2 Bytes	Identification of a MODBUS Request / Response transaction.	Initialized by the client	Recopied by the server from the received request
Protocol Identifier	2 Bytes	0 = MODBUS protocol	Initialized by the client	Recopied by the server from the received request
Length	2 Bytes	Number of following bytes	Initialized by the client ( request)	Initialized by the server ( Response)
Unit Identifier	1 Byte	Identification of a remote slave connected on a serial line or on other buses.	Initialized by the client	Recopied by the server from the received request

**MBAP Header**

### 3 支持命令

目前，本公司所生产的以太网分布式采集模块均采用该协议，MODBUS TCP 方式。支持的功能码主要包括如下几种：

- 01 READ COIL STATUS
- 02 READ INPUT STATUS
- 03 READ HOLDING REGISTERS
- 04 READ INPUT REGISTERS
- 05 FORCE SINGLE COIL
- 06 PRESET SINGLE REGISTER
- 15 FORCE MULTIPLE COILS
- 16 FORCE MULTIPLE REGISTERS
- 20 READ FILE RECORD
- 21 WRITE FILE RECORD

### 4 地址映射表

#### 4.1 读开关量输入

功能码：02

说明：读取输入开关量的状态

数据说明：

地址	描述	说明
10001	第 01 路开关量输入状态	=0 没有通电 =1 接通电源
10002	第 02 路开关量输入状态	=0 没有通电 =1 接通电源
10003	第 03 路开关量输入状态	=0 没有通电 =1 接通电源
10004	第 04 路开关量输入状态	=0 没有通电 =1 接通电源
10005	第 05 路开关量输入状态	=0 没有通电 =1 接通电源
10006	第 06 路开关量输入状态	=0 没有通电 =1 接通电源
10007	第 07 路开关量输入状态	=0 没有通电 =1 接通电源
10008	第 08 路开关量输入状态	=0 没有通电 =1 接通电源
10009	第 09 路开关量输入状态	=0 没有通电 =1 接通电源
10010	第 10 路开关量输入状态	=0 没有通电 =1 接通电源
10011	第 11 路开关量输入状态	=0 没有通电 =1 接通电源
10012	第 12 路开关量输入状态	=0 没有通电 =1 接通电源
10013	第 13 路开关量输入状态	=0 没有通电 =1 接通电源
10014	第 14 路开关量输入状态	=0 没有通电 =1 接通电源
10015	第 15 路开关量输入状态	=0 没有通电 =1 接通电源
10016	第 16 路开关量输入状态	=0 没有通电 =1 接通电源
保 留		
10033	第 01 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10034	第 02 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10035	第 03 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10036	第 04 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10037	第 05 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10038	第 06 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10039	第 07 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10040	第 08 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10041	第 09 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10042	第 10 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10043	第 11 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10044	第 12 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10045	第 13 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10046	第 14 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10047	第 15 路上升沿锁存状态	=0 没有锁存 =1 有锁存
10048	第 16 路上升沿锁存状态	=0 没有锁存 =1 有锁存
保 留		
10065	第 01 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10066	第 02 路下降沿锁存状态	=0 没有锁存 =1 有锁存

10067	第 03 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10068	第 04 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10069	第 05 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10070	第 06 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10071	第 07 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10072	第 08 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10073	第 09 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10074	第 10 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10075	第 11 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10076	第 12 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10077	第 13 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10078	第 14 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10079	第 15 路下降沿锁存状态	=0 没有锁存 =1 有锁存
10080	第 16 路下降沿锁存状态	=0 没有锁存 =1 有锁存

### MODBUS 请求

域名称	字节数	取值
功能码	1byte	0x02
起始地址	2byte	0x0000 to 0xFFFF
读取数量	2byte	1 to 2000(0x7D0)

### MODBUS 响应

域名称	字节数	取值
功能码	1byte	0x02
字节计数	1byte	$n = (\text{读取数量} + 7) / 8$
输入状态	nbyte	

### 错误响应

域名称	字节数	取值
功能码	1byte	0x02+0x80
错误代码	1byte	0x1 or 0x2

### 举例说明

请求		响应	
域名称	数据 (hex)	域名称	数据 (hex)
功能码	02	功能码	02
起始地址高(字节)	00	字节计数	03
起始地址低(字节)	C4	204(h)~197 状态	AC
读取数量高(字节)	00	212(h)~205 状态	DB
读取数量低(字节)	16	218(h)~213 状态	35



## 4.2 读保持寄存器

功能码：03

说明：读取保持寄存器的值

数据说明：读取的是十六位整数或无符号整数

地址	描述	说明
40065	第 01 路开关量脉冲计数	低 16 位计数数值
40066	第 01 路开关量脉冲计数	高 16 位计数数值
40067	第 02 路开关量脉冲计数	低 16 位计数数值
40068	第 02 路开关量脉冲计数	高 16 位计数数值
40069	第 03 路开关量脉冲计数	低 16 位计数数值
40070	第 03 路开关量脉冲计数	高 16 位计数数值
40071	第 04 路开关量脉冲计数	低 16 位计数数值
40072	第 04 路开关量脉冲计数	高 16 位计数数值
40073	第 05 路开关量脉冲计数	低 16 位计数数值
40074	第 05 路开关量脉冲计数	高 16 位计数数值
40075	第 06 路开关量脉冲计数	低 16 位计数数值
40076	第 06 路开关量脉冲计数	高 16 位计数数值
40077	第 07 路开关量脉冲计数	低 16 位计数数值
40078	第 07 路开关量脉冲计数	高 16 位计数数值
40079	第 08 路开关量脉冲计数	低 16 位计数数值
40080	第 08 路开关量脉冲计数	高 16 位计数数值
40081	第 09 路开关量脉冲计数	低 16 位计数数值
40082	第 09 路开关量脉冲计数	高 16 位计数数值
40083	第 10 路开关量脉冲计数	低 16 位计数数值
40084	第 10 路开关量脉冲计数	高 16 位计数数值
40085	第 11 路开关量脉冲计数	低 16 位计数数值
40086	第 11 路开关量脉冲计数	高 16 位计数数值
40087	第 12 路开关量脉冲计数	低 16 位计数数值
40088	第 12 路开关量脉冲计数	高 16 位计数数值
40089	第 13 路开关量脉冲计数	低 16 位计数数值
40090	第 13 路开关量脉冲计数	高 16 位计数数值
40091	第 14 路开关量脉冲计数	低 16 位计数数值
40092	第 14 路开关量脉冲计数	高 16 位计数数值
40093	第 15 路开关量脉冲计数	低 16 位计数数值
40094	第 15 路开关量脉冲计数	高 16 位计数数值
40095	第 16 路开关量脉冲计数	低 16 位计数数值
40096	第 16 路开关量脉冲计数	高 16 位计数数值
保留		
40130	输入锁存使能寄存器	置位：开启，清位：禁止

40131	输入计数使能寄存器	置位：开启，清位：禁止
40132	输入计数模式寄存器	置位：上升，清位：下降
保留		
40513	看门狗控制寄存器	bit0: 0 禁止, 1 开启 bit1: 0 正常, 1 溢出 bit2: 0 正常, 1 复位
40514	看门狗溢出寄存器	看门狗超时时间常数
40515	看门狗复位寄存器	0x55AA
保留		

### MODBUS 请求

域名称	字节数	取值
功能码	1byte	0x03
起始地址	2byte	0x0000 to 0xFFFF
读取数量	2byte	1 to 125(0x7D)

### MODBUS 响应

域名称	字节数	取值
功能码	1byte	0x03
字节计数	1byte	2n
输入状态	2nbyte	

### 错误响应

域名称	字节数	取值
功能码	1byte	0x03+0x80
错误代码	1byte	0x1 or 0x2

### 举例说明

请求		响应	
域名称	数据 (hex)	域名称	数据 (hex)
功能码	03	功能码	03
起始地址高(字节)	00	字节计数	02
起始地址低(字节)	08	输入寄存器高	00
读取数量高(字节)	00	输入寄存器低	0A
读取数量低(字节)	01		

注 1: 脉冲输出电平宽度单位是：毫秒；看门狗定时长度单位是：毫秒

注 2: 看门狗控制寄存器的最高位上电为 1，可以做模块复位判断。

## 4.3 设置单个继电器

功能码：05

**MODBUS 请求**

域名称	字节数	取值
功能码	1byte	0x05
设置地址	2byte	0x0000 to 0xFFFF
设置内容	2byte	0x0000 or 0xFF00 0x0000 释放继电器 0xFF00 吸合继电器

**MODBUS 响应**

域名称	字节数	取值
功能码	1byte	0x05
设置地址	2byte	0x0000 to 0xFFFF
设置内容	2byte	0x0000 or 0xFF00

**错误响应**

域名称	字节数	取值
功能码	1byte	0x05+0x80
错误代码	1byte	0x1 or 0x2

**举例说明**

请求		响应	
域名称	数据 (hex)	域名称	数据 (hex)
功能码	05	功能码	05
设置地址高(字节)	00	设置地址高(字节)	00
设置地址低(字节)	05	设置地址低(字节)	05
设置内容高(字节)	FF	设置内容高(字节)	FF
设置内容低(字节)	00	设置内容低(字节)	00

**4.4 设置单个保持寄存器****功能码: 06****MODBUS 请求**

域名称	字节数	取值
功能码	1byte	0x06
设置地址	2byte	0x0000 to 0xFFFF
设置内容	2byte	0x0000 to 0xFFFF

**MODBUS 响应**

域名称	字节数	取值
功能码	1byte	0x06

设置地址	2byte	0x0000 to 0xFFFF
设置内容	2byte	0x0000 to 0xFFFF

### 错误响应

域名称	字节数	取值
功能码	1byte	0x06+0x80
错误代码	1byte	0x1 or 0x2

### 举例说明

请求		响应	
域名称	数据 (hex)	域名称	数据 (hex)
功能码	06	功能码	06
设置地址高(字节)	00	设置地址高(字节)	00
设置地址低(字节)	08	设置地址低(字节)	08
设置内容高(字节)	00	设置内容高(字节)	00
设置内容低(字节)	19	设置内容低(字节)	19

## 4.5 设置多个继电器

功能码：0F

### MODBUS 请求

域名称	字节数	取值
功能码	1byte	0x0F
设置起始地址	2byte	0x0000 to 0xFFFF
设置长度	2byte	0x0000 to 0x07B0
字节计数	1byte	n
设置内容	nbyte	

### MODBUS 响应

域名称	字节数	取值
功能码	1byte	0x0F
设置起始地址	2byte	0x0000 to 0xFFFF
设置长度	2byte	0x0000 to 0x07B0

### 错误响应

域名称	字节数	取值
功能码	1byte	0x0F+0x80
错误代码	1byte	0x1 or 0x2

### 举例说明

请求		响应	
域名称	数据 (hex)	域名称	数据 (hex)
功能码	0F	功能码	0F
设置地址高(字节)	00	设置地址高(字节)	00
设置地址低(字节)	13	设置地址低(字节)	13
设置数量高(字节)	00	设置数量高(字节)	00
设置数量低(字节)	0A	设置数量低(字节)	0A
字节计数	02		
设置内容高(字节)	CD		
设置内容低(字节)	01		

## 4.6 设置多个保持寄存器

功能码：10

### MODBUS 请求

域名称	字节数	取值
功能码	1byte	0x10
设置起始地址	2byte	0x0000 to 0xFFFF
设置长度	2byte	0x0000 to 0x07B0
字节计数	1byte	2n
设置内容	2nbyte	

### MODBUS 响应

域名称	字节数	取值
功能码	1byte	0x10
设置起始地址	2byte	0x0000 to 0xFFFF
设置长度	2byte	0x0000 to 0x07B0

### 错误响应

域名称	字节数	取值
功能码	1byte	0x10+0x80
错误代码	1byte	0x1 or 0x2

### 举例说明

请求		响应	
域名称	数据 (hex)	域名称	数据 (hex)
功能码	10	功能码	10
设置地址高(字节)	00	设置地址高(字节)	00
设置地址低(字节)	01	设置地址低(字节)	01
设置数量高(字节)	00	设置数量高(字节)	00
设置数量低(字节)	02	设置数量低(字节)	02

字节计数	04	
设置内容高(字节)	00	
设置内容低(字节)	0A	
设置内容高(字节)	01	
设置内容低(字节)	02	

## 4.7 读文件记录

### 功能码：14/06

读取文件记录，在 MODBUS 中，认为文件是一个由 16BIT 位串构成的数组，其寻址是按照地址进行的。文件读取，规定读取的起始地址和读取长度，改变读取地址和长度就可以遍历整个文件。文件没有名字，只有编号。本系统仅支持一次读写一个文件。

### MODBUS 请求

域名称	字节数	取值
功能码	1byte	0x14
字节计数	1byte	0x07 to 0xF5
子功能码	1byte	0x06
文件号	2byte	0x0000 to 0xFFFF
记录号	2byte	0x0000 to 0x270F
读取长度	2byte	n
子功能码	1byte	0x06
.....	.....	

### MODBUS 响应

域名称	字节数	取值
功能码	1byte	0x14
字节计数	1byte	0x07 to 0xF5
子功能字节计数	1byte	0x07 to 0xF5
子功能码	1byte	0x06
数据	2nbyte	

### 错误响应

域名称	字节数	取值
功能码	1byte	0x14+0x80
错误代码	1byte	0x1 or 0x2

### 举例说明

请求		响应	
域名称	数据 (hex)	域名称	数据 (hex)
功能码	14	功能码	14

字节计数	07	字节计数	06
子功能码	06	响应计数	05
文件号高(字节)	00	子功能码	06
文件号低(字节)	04	记录数据高(字节)	0D
记录号高(字节)	00	记录数据低(字节)	FE
记录号低(字节)	01	记录数据高(字节)	00
读取长度高(字节)	00	记录数据低(字节)	20
读取长度低(字节)	02		

## 4.8 写文件记录

功能码：15/06

### MODBUS 请求

域名称	字节数	取值
功能码	1byte	0x15
字节计数	1byte	0x07 to 0xF5
子功能码	1byte	0x06
文件号	2byte	0x0000 to 0xFFFF
记录号	2byte	0x0000 to 0x270F
写长度	2byte	n
数据	2nbyte	
.....	.....	

### MODBUS 响应

域名称	字节数	取值
功能码	1byte	0x15
字节计数	1byte	0x07 to 0xF5
子功能码	1byte	0x06
文件号	2byte	0x0000 to 0xFFFF
记录号	2byte	0x0000 to 0x270F
写长度	2byte	n
数据	2nbyte	

### 错误响应

域名称	字节数	取值
功能码	1byte	0x15+0x80
错误代码	1byte	0x1 or 0x2

### 举例说明

请求		响应	
域名称	数据 (hex)	域名称	数据 (hex)

功能码	15	功能码	15
字节计数	0B	字节计数	0B
子功能码	06	子功能码	06
文件号高(字节)	00	文件号高(字节)	00
文件号低(字节)	04	文件号低(字节)	04
记录号高(字节)	00	记录号高(字节)	00
记录号低(字节)	01	记录号低(字节)	01
写长度高(字节)	00	写长度高(字节)	00
写长度低(字节)	02	写长度低(字节)	02
写数据	4byte	写数据	4byte

## 4.9 EEPROM 分配

//file 0

```
#define MODULE_NET_ADDR 0x0000
```

```
#define MODULE_VER_ADDR 0x0020
```

//file 1

```
#define NET_CONFIG_ADDR 0x0000
```

//file 2to file 7

.....

### 说明:

EEPROM 为 8kbyte 容量，分为 8 个文件块，每个大小为 1kbyte。

#### 1、MODULE\_NET\_ADDR

恢复出厂设置的网络配置参数。结构如下表:

字节数	4	4	4	6
内容	IP 地址	默认网关	子网掩码	MAC 地址

#### 2、MODULE\_VER\_ADDR

模块版本信息。结构如下表:

字节	42byte
内容	DAM-E3016 V6.20 2006.09.01 ID:DAME123456

#### 3、NET\_CONFIG\_ADDR

网络配置参数。结构如上表 1。