

---

NA5 系列万能式断路器通讯协议 (M 型智能控制器)

# 使用手册



浙江正泰电器股份有限公司

2022 年 11 月

# 目 录

0	前言 .....	
1	范围 .....	(1)
2	术语和定义 .....	(1)
2.1	开发系统互连模型 .....	(1)
2.2	物理层 .....	(1)
2.3	数据链路层 .....	(1)
2.4	应用层 .....	(1)
2.5	帧 .....	(1)
3	Modbus 协议简介 .....	(1)
4	协议概述 .....	(2)
4.1	物理层 .....	(2)
4.2	数据链路层 .....	(2)
4.2.1	传输方式 .....	(2)
4.2.2	协议类型 .....	(2)
4.2.3	串行传输格式 .....	(2)
4.2.4	数据包格式 .....	(2)
4.3	应用层 .....	(2)
4.3.1	地址码 .....	(3)
4.3.2	功能码 .....	(3)
4.3.3	数据域 .....	(3)
4.3.4	校验码 .....	(3)
4.3.5	应用层功能详解 .....	(3)
4.3.5.1	读数据寄存器 (03H) .....	(3)
4.3.5.2	写数据寄存器 (06H) .....	(4)
5	通讯数据表 .....	(5)
5.1	测量信息 .....	(6)
5.2	系统维护 .....	(6)
5.3	故障查询 .....	(6)
5.4	参数设置 .....	(7)
5.5	远程控制 .....	(7)
附表 A	.....	(7)
附表 B	.....	(8)
附表 C	.....	(8)
附表 E	.....	(8)
附表 F	.....	(8)
附表 G	.....	(9)
附 F1	CRC-16 生成式原理 .....	(9)

## 前 言

本使用手册修改采用 GB/T 27745-2011 《低压电器通信规范》。

本使用手册由浙江正泰电器股份有限公司配电电器制造二部提出。

本使用手册仅代表本次版本的内容，如有更新，将不作通知，请关注我公司最新版本。

文档记录:

版本	修改内容	日期
V1.0	新建文档	2020/06
V1.1	a. 修改脱扣信息 (L1 脱扣电流->第一项脱扣数据), 并添加脱扣数据格式说明表 (见附表 E); b. 新增文档记录; c. 去除内容相同的注解; d. 长延时/短延时/瞬时/接地保护关闭方式更改 (时间整定值->电流整定值);	2022/9
V1.2	a. 添加 ModbusRTU 格式和 ModbusTCP 报文格式说明; b. 添加 ModbusTCP 示例; c. 去除 03H/06H 访问规则, 改为 03H/06H;	2022/11

## NA5 系列万能式断路器 M 型智能控制器通讯协议使用手册

### 1 范围

本使用手册规定了 Modbus 协议的基本术语、协议组成和通讯数据表。

本使用手册适用于 NA5 系列万能式断路器 M 型智能控制器选配通讯功能产品。

### 2 术语和定义

下列术语适用于本使用手册。

#### 2.1 开放系统互连 (OSI) 模型

国际标准化组织 (ISO) 于 1984 年制定的标准, 目的是为不同厂家的计算机能互连提供一个共同的基础和标准框架。

#### 2.2 物理层

在开放系统互连 (OSI) 模型中的第一层, 为通信提供实现透明传输的物理链接。

#### 2.3 数据链路层

在开放系统互连 (OSI) 模型中的第二层, 提供相邻节点间透明、可靠的信息传输服务。

#### 2.4 应用层

在开放系统互连 (OSI) 模型中的第七层, 实现数据操作和信息交换的具体功能。

#### 2.5 帧

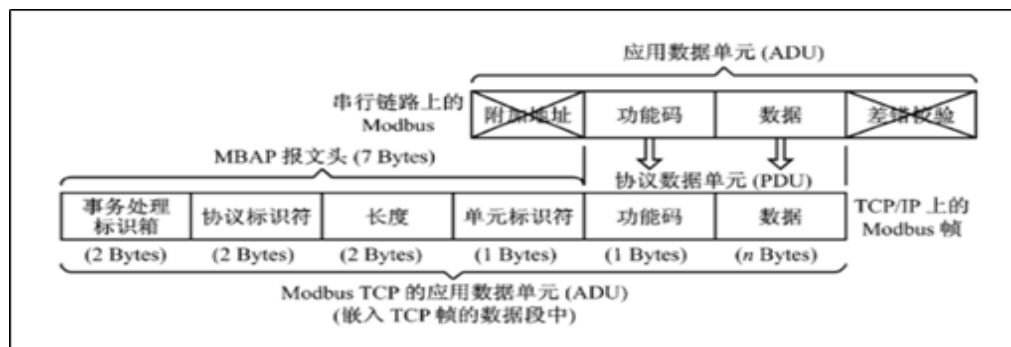
数据和数字通信中, 按某一标准预先确定的若干比特或字段组成的特定的信息结构。数据在网络上是以很小的称为帧 (Frame) 的单位传输的, 帧由几部分组成, 不同的部分执行不同的功能。

### 3 Modbus 协议简介

Modbus 协议是一种基于 ISO/OSI 模型 (7 层) 设计的工业总线协议, 但只选取了 7 层结构中的 3 层 (物理层、数据链路层和应用层) 进行使用, 简化了协议模型, 降低了使用难度。

我公司生产的断路器采用 RTU 方式, 如需 Modbus-TCP 协议通讯方式, 请使用我公司“COMA-A5E 通讯模块”。

#### 3.1 ModbusRTU 与 ModbusTCP 区别:



#### 3.2 ModbusTCP 字段说明:

从上图可以看出, ModbusTCP 在 Modbus 串行通信的基础上, 去除了校验(由于 TCP 本身就带有校验和)和设备地址( ModbusTCP 弱化了设备地址, 用 IP 地址来取代), 再加上 MBAP 报文头(占 7 bytes), 下面针对 MBAP 部分进行分析说明:

域	长度	说明	客户机	服务器
事务处理标识符	2 字节	Modbus 请求/响应事务处理的报文标识, 一般每次通讯之后就要加一, 以区分不同的通讯报文。	客户机启动	复制响应
协议标识符	2 字节	0=Modbus 协议	客户机启动	复制响应
长度	2 字节	长度之后的字节总数	客户机启动	服务器启动
单元标识符	1 字节	串行链路或其它总线的从站识别	客户机启动	复制响应

下面针对具体报文进行分析, Modbus 协议在以太网链路上的报文格式如下所示:

事务处理标识符	协议标识符	长度	单元标识符	功能码	数据
2 字节	2 字节	2 字节	1 字节	1 字节	N 字节

注意: 具体示例, 请见下面 RTU 报文格式/TCP 报文格式。

## 4 协议概述

### 4.1 物理层

物理层参数	物理层内容	备注
通讯方式	RS-485	半双工
通讯地址	3 ~ 247 可选	默认: 3
通讯波特率	9.6kbps	
通讯距离	≤ 1000 m	低波特率时
通讯介质	屏蔽双绞线	A 类
最大连接数量	最多 32 台	多机组网时

### 4.2 数据链路层

4.2.1 传输方式: 采用主从半双工方式。(主机查询, 从机应答)

4.2.2 协议类型: 通讯协议采用 Modbus - RTU 方式。

4.2.3 串行传输格式: 1 起始位, 8 数据位, 无校验位, 默认 2 停止位。(1 帧数据)

起始	数 据								停 止	
Start	1	2	3	4	5	6	7	8	Stop	Stop

4.2.4 数据包(多帧)格式: 如表

开 始	地址帧	功能帧	数据帧	校验帧	终 止
T3.5	8 bits	8 bits	n × 8 bits	16 bits	T3.5

注: RTU 模式中, 信息开始至少需要要有 3.5 个字符(或帧)的静止时间, 依据使用的波特率, 很容易就算出这个静止的时间(如上表中 T3.5)。这个延时对使用单片机的 UART 来制作通讯协议时要考虑, 如果采用组态软件或 DCS 使用时无需考虑, 软件底层已做好。

### 4.3 应用层

应用层是对数据包(包括地址码, 功能码, 数据域, 校验码等)进行解析, 达到数据交换的目的。

当主机发送的数据包到达从机设备时, 它通过通讯端口进入寻址到的设备, 该从机设备去掉数据包的“信封”(数据头), 读取有效数据; 如果数据没有错误, 就执行数据所请求的任务, 并将自己生成的数据加入到取得的“信封”中, 形成新的数据包, 返回给主机。返回

的响应数据中包含了以下内容：从机地址(Address)、被执行了的命令(Function)、执行命令生成的被请求数据(Data)和一个校验码(Check)。

#### 4.3.1 地址码

地址码在帧的开始部分，由 8 位组成（取 3~247），这些位标明了用户指定的从机设备的地址，该从机设备将接收来自与之相连的主机数据。在同一网络中每个从机设备的地址必须是唯一的，仅仅被寻址到的从机会响应包含了该地址的查询。当从机发送回一个响应，响应中的从机地址数据便告诉了主机是哪台从机正与之进行通讯。

#### 4.3.2 功能码

功能码告诉了被寻址到的从机执行何种功能。所有的功能码、定义和行为见下表。

功能码	定 义	功 能 行 为
03H	读数据寄存器	读取一个或多个寄存器的数据值
06H	写单个寄存器	写入数据到一个寄存器

#### 4.3.3 数据域

数据域包含了从机执行特定功能所需要的数据或者从机响应查询时采集到的数据。这些数据的内容可能是数值、参考地址或者极限值，具体内容参照通讯地址寄存器表。例如：功能域码告诉从机读取一个寄存器，数据域则需要指明从哪个寄存器开始及读取多少个数据，内嵌的地址和数据依照类型和从机之间的不同能力而有所不同。

#### 4.3.4 校验码

该域允许主机和从机检查传输过程中的错误。有时，由于电噪声和其它干扰，一组数据在从一个设备传输到另一个设备时在线路上可能会发生一些改变，出错校验能够保证主机或者从机不去响应那些传输过程中发生了改变的数据，这就提高了系统的安全性和效率，出错校验使用了 16 位循环冗余的方法。

循环冗余校验（CRC）域占用两个字节，包含了一个 16 位的二进制值。CRC 值由传送设备计算出来，然后附加到数据帧上，接收设备在接收数据时重新计算 CRC 值，然后与接收到的 CRC 域中的值进行比较，如果这两个值不相等，就发生了错误。

注：CRC16 的生成方法见附 F1 CRC-16 生成式原理。

#### 4.3.5 应用层功能详解

##### 4.3.5.1 读数据寄存器（03H）

03H 功能码允许用户获得智能控制器采集与记录的数据及系统参数。

例如：读取 Ia, Ib, Ic 三相电流值，根据返回数据其结果为 Ia=0001, Ib=0002, Ic=0003。

RTU 报文格式示例						
主机查询			从机应答			
帧 域	内容	说 明	帧 域	内容	说 明	
地址码	03	从机地址	地址码	03	从机地址	
功能码	03	功能码	功能码	03	功能码	
数据域	00	读取寄存器地址高字节	数据域	06	返回数据总个数	
	00	读取寄存器地址低字节		00	数据 1 高字节	
				01	数据 1 低字节	
				00	数据 2 高字节	

	00	读取数据个数高字节		02	数据 2 低字节
	03	读取数据个数低字节		00	数据 3 高字节
校验码	04	CRC 校验低字节	校验码	03	数据 3 低字节
	29	CRC 校验高字节		E4	CRC 校验低字节
				14	CRC 校验高字节

主机发送 [ 03 03 00 00 00 03 04 29 ]

从机响应 [ 03 03 06 00 01 00 02 00 03 e4 14 ]

TCP 报文格式示例					
主机查询			从机应答		
帧域	内容	说明	帧域	内容	说明
事务标识	00	第一次通讯报文	事务标识	00	复制响应
	00			00	
协议标识	00	0 代表 Modbus 协议 (固定)	协议标识	00	复制响应
	00			00	
长度	00	长度之后的字节总数	长度	00	长度之后的字节总数
	06			09	
单元标识	03	服务器从站地址	单元标识	03	服务器从站地址
功能码	03	03H 读指令	功能码	03	复制响应
数据域	00	读取寄存器地址高字节	字节计数	06	返回数据总个数
	00	读取寄存器地址低字节			
	00	读取数据个数高字节	数据域	00	数据 1 高字节
	03	读取数据个数低字节		01	数据 1 低字节
				00	数据 2 高字节
				02	数据 2 低字节
				00	数据 3 高字节
				03	数据 3 低字节

主机发送 [ 00 00 00 00 00 05 03 00 00 00 03 ]

从机响应 [ 00 00 00 00 00 09 03 03 06 00 01 00 02 00 03 ]

#### 4.3.5.2 写数据寄存器 (06H)

功能码 06H 允许用户修改单个寄存器的内容, 智能控制器内部的任何可写的寄存器都可以使用此功能码来改变其值。

例如: 将长延时电流整定值(寄存器地址为 0x2007)修改为 2500A(十六进制为 0x09C4)。

RTU 报文格式示例					
主机查询			从机应答		
帧域	内容	说明	帧域	内容	说明
地址码	03	从机地址	地址码	03	从机地址
功能码	06	功能码	功能码	06	功能码

数据域	20	待写入地址高字节	数据域	20	写入地址高字节
	07	待写入地址高字节		07	写入地址高字节
	09	写入数据高字节		09	写入数据高字节
	C4	写入数据低字节		C4	写入数据低字节
校验码	35	CRC 校验低字节	校验码	35	CRC 校验低字节
	EA	CRC 校验高字节		EA	CRC 校验高字节

主机发送 [ 03 06 20 07 09 C4 35 EA ]

从机响应 [ 03 06 20 07 09 C4 35 EA ]

TCP 报文格式示例					
主机查询			从机应答		
帧域	内容	说明	帧域	内容	说明
事务标识	00	第一次通讯报文	事务标识	00	复制响应
	00			00	
协议标识	00	0 代表 Modbus 协议 (固定)	协议标识	00	复制响应
	00			00	
长度	00	长度之后的字节总数	长度	00	长度之后的字节总数
	06			06	
单元标识	03	服务器从站地址	单元标识	03	服务器从站地址
功能码	06	06H 写指令	功能码	06	复制响应
数据域	20	待写入地址高字节	数据域	20	写入地址高字节
	07	待写入地址高字节		07	写入地址高字节
	09	写入数据高字节		09	写入数据高字节
	C4	写入数据低字节		C4	写入数据低字节

主机发送 [00 00 00 00 00 06 03 06 20 07 09 C4]

从机响应 [00 00 00 00 00 06 03 06 20 07 09 C4]

5 通讯数据表

数据类型:

SINT8 表示有符号 单字节参数;

UINT8 表示无符号 单字节参数;

SINT16 表示有符号 双字节参数;

UINT16 表示无符号双字节参数;

SINT32 表示有符号 4 字节参数;

UINT32 表示无符号 4 字节参数;

其中: 单个字节为 Byte, 共 8 个 bit。

数据高低位:

UINT8 单字节参数举例:

Bit0: 为最低位, 用 L 表示。(L 意为“低”, 英文 Low 的缩写);

Bit7: 为最高位, 用 H 表示。(H 意为“高”, 英文 High 的缩写);

Bit	Bit	Bit	Bit	Bit	Bit	Bit	Bit
-----	-----	-----	-----	-----	-----	-----	-----



7	6	5	4	3	2	1	0
H(高 4bit)				L(低 4bit)			

UINT16 双字节参数举例:

Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
HH(最高 4bit)				HL(次高 4bit)				LH(次低 4bit)				LL(最低 4bit)			

UINT32 四字节参数举例:

Byte31~Byte24				Byte23~Byte16				Byte15~Byte8				Byte7~Byte0			
HH(最高 8bit)				HL(次高 8bit)				LH(次低 8bit)				LL(最低 8bit)			

### 5.1 测量信息

序号	参 数 项	数据类型	单位	访问规则	地 址	备 注
1	工作状态字	UINT16	—	03H	0x0000	见附表 A
2	L1 电流	UINT16	A	03H	0x0001	注 1
3	L2 电流	UINT16	A	03H	0x0002	注 1
4	L3 电流	UINT16	A	03H	0x0003	注 1
5	LN 电流	UINT16	A	03H	0x0004	注 1
6	LG 电流	UINT16	A	03H	0x0005	注 1
7	电流不平衡率(最大)	UINT16	%	03H	0x0011	

注 1: 壳架电流 (或框架电流)  $\leq 2500$  时  $\times 1$ , 其他壳架电流时  $\times 2$ 。

### 5.2 系统维护

序号	参 数 项	数据类型	单位	访问规则	地 址	备 注
1	ModBus 地址	UINT16	—	03H/06H	0x0100	
2	ModBus 波特率	UINT16	—	03H	0x0101	
3	控制器类型	UINT16		03H	0x0143	见附表 G
4	壳架电流	UINT16	A	03H	0x0180	见附表 C
5	额定电流	UINT16	A	03H	0x0181	注 1

### 5.3 故障查询

序号	参 数 项	数据类型	单位	访问规则	地 址	备 注
1	第 1 次脱扣原因	UINT16	—	03H	0x0203	见附表 B
2	第 1 次脱扣数据	UINT16	A	03H	0x0208	见附表 E
3	第 1 次脱扣动作时间	UINT16	s	03H	0x0213	$\times 0.01$
4	第 2 次脱扣原因	UINT16		03H	0x0217	见附表 B
5	第 2 次脱扣数据	UINT16	A	03H	0x021C	见附表 E
6	第 2 次脱扣动作时间	UINT16	s	03H	0x0227	$\times 0.01$
7	第 3 次脱扣原因	UINT16		03H	0x022B	见附表 B
8	第 3 次脱扣动作时间	UINT16	s	03H	0x0230	$\times 0.01$
9	第 4 次脱扣原因	UINT16		03H	0x0234	见附表 B
10	第 4 次脱扣动作时间	UINT16	s	03H	0x0239	$\times 0.01$
11	第 5 次脱扣原因	UINT16		03H	0x023D	见附表 B

12	第 5 次脱扣动作时间	UINT16	s	03H	0x0242	× 0.01
13	第 6 次脱扣原因	UINT16		03H	0x0246	见附表 B
14	第 6 次脱扣动作时间	UINT16	s	03H	0x024B	× 0.01
15	第 7 次脱扣原因	UINT16		03H	0x024F	见附表 B
16	第 7 次脱扣动作时间	UINT16	s	03H	0x0254	× 0.01
17	第 8 次脱扣原因	UINT16		03H	0x0258	见附表 B
18	第 8 次脱扣动作时间	UINT16	s	03H	0x025D	× 0.01
19	第 9 次脱扣原因	UINT16		03H	0x0261	见附表 B
20	第 9 次脱扣动作时间	UINT16	s	03H	0x0262	× 0.01
21	第 10 次脱扣原因	UINT16		03H	0x0266	见附表 B
22	第 10 次脱扣动作时间	UINT16	s	03H	0x0267	× 0.01

## 5.4 参数设置

序号	参 数 项	数据类型	单位	访问规则	地 址	备 注
1	长延时电流整定值	UINT16	A	03H/06H	0x2007	注 1、注 6、注 7
2	长延时时间整定值	UINT16	s	03H/06H	0x2008	注 6
3	短延时电流整定值	UINT16	A	03H/06H	0x2009	注 1、注 6、注 7
4	短延时时间整定值	UINT16	ms	03H/06H	0x200A	注 6
5	瞬时电流整定值	UINT16	A	03H/06H	0x200B	注 1、注 6、注 7
6	接地保护电流整定值	UINT16	A	03H/06H	0x200C	注 1、注 7
7	接地保护时间整定值	UINT16	ms	03H/06H	0x200D	注 6、
8	电流不平衡动作阈值整定值	UINT16	%	03H/06H	0x200E	
9	电流不平衡动作时间整定值	UINT16	s	03H/06H	0x200F	注 7
10	电流断相动作阈值整定值	UINT16	%	03H/06H	0x2012	
11	电流断相动作时间整定值	UINT16	s	03H/06H	0x2013	× 0.1, 注 7

注 6: 参数设置范围见说明书, 当保护电流整定值为 65535 或者 0xFFFF 时, 意为关闭此保护, 当电流整定值不为 65535 或者 0xFFFF, 意为开启此保护。

注 7: 数据为 65535 (0xFFFF) 时, 表示关闭该功能。

## 5.5 远程控制

序号	参 数 项	数据类型	单位	访问规则	地 址	备 注
1	控制命令	UINT16	—	06H	0x2800	见表 F

注 8: 遥控分合闸需配备 PSU-A52 电源模块及 RU-A5 继电器模块。

附表 A 工作状态字位格式

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
—	—	—	—	—	—	—	—
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
—	—	开关分	故障脱				

		合指示 状态位 0: 合 1: 分	扣指示 状态位 0: 正常 1: 故障			
--	--	----------------------------	------------------------------	--	--	--

附表 B 脱扣原因数据说明表

脱扣原因代号	脱扣类型名称	备 注
00H(十六进制)	无脱扣故障	0(十进制)
01H	过载长延时故障	1
02H	短路短延时故障	2
03H	短路瞬时故障	3
04H	接地故障	4
07H	电流不平衡故障	7
08H	电流断相故障	8

附表 C 壳架电流数据说明表

壳架电流代号	壳架电流	备 注
00H(十六进制)	1000 壳架	0(十进制)
01H	1600 壳架	1
02H	2000 壳架	2
03H	2500 壳架	3
04H	3200 壳架	4
05H	4000 壳架	5
06H	6300 壳架	6
07H	7500 壳架	7

附表 E 脱扣数据格式(随脱扣原因改变)

脱扣原因	脱扣数据格式	单位	备注
过载长延时故障	故障电流	A	注 1
短路短延时故障	故障电流	A	注 1
短路瞬时故障	故障电流	A	注 1
接地故障	故障电流	A	注 1
矢量和故障	故障电流	A	注 1
中性极故障	故障电流	A	注 1
电流不平衡故障	最大不平衡率	%	
电流断相故障	最大不平衡率	%	
其他故障	无	无	无

附表 F 控制命令位格式

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
-	-			-	-	开关分合闸控制位 01: 分闸 10: 合闸	

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
—	—	—	—	—	—	—	—

附表 G 控制器类型格式

Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
无							
Bit7 ~ Bit0 控制器类型							
NA5-M 型控制器: 0x01							

## 附 F1 CRC-16 生成式原理

CRC 运算时, 首先将一个 16 位的寄存器预置为全 1, 然后连续把数据帧中的 8 位字节与该寄存器的当前值进行运算, 仅仅每个字节的 8 个数据位参与生成 CRC, 起始位和终止位以及可能使用的奇偶位都不影响 CRC。

在生成 CRC 时, 每个 8 位字节与寄存器中的内容进行异或, 然后将结果向低位移位, 高位则用“0”补充, 最低位 (LSB) 移出并检测, 如果是 1, 该寄存器就与一个预设的固定值进行一次异或运算, 如果最低位为 0, 不作任何处理。

上述处理重复进行, 直到执行完了 8 次移位操作, 当最后一位 (第 8 位) 移完以后, 下一个 8 位字节与寄存器材的当前值进行异或运算, 同样进行上述的另一个 8 次移位异或操作, 当数据帧中的所有字节都作了处理, 生成的最终值就是 CRC 值。

生成一个 CRC 的流程为:

- 预置一个 16 位寄存器为 0FFFFH (全 1), 称之为 CRC 寄存器;
- 把数据帧中的第一个 8 位字节与 CRC 寄存器中的低字节进行异或运算, 结果存回 CRC 寄存器;
- 将 CRC 寄存器向右移一位, 最高位填以 0, 最低位移出并检测;
- 如果最低位为 0: 重复第三步 (下一次移位);
- 如果最低位为 1: 将 CRC 寄存器与一个预设的固定值 (0A001H) 进行异或运算;
- 重复第三步和第四步直到 8 次移位。这样处理完了一个完整的八位;
- 重复第 2 步到第 5 步来处理下一个八位, 直到所有的字节处理结束;
- 最终 CRC 寄存器得值就是 CRC 的值。