

NXZ (H) M 、NXZB自动转换开关

控制器通讯协议

Modbus-RTU

使用手册

浙江正泰电器股份有限公司

2018年1月

目录

- 0 前言
 - 1 范围
 - 2 术语和定义
 - 2.1 开发系统互连模型
 - 2.2 物理层
 - 2.3 数据链路层
 - 2.4 应用层
 - 2.5 帧
 - 3 Modbus 协议简介
 - 4 协议概述
 - 4.1 物理层
 - 4.2 数据链路层
 - 4.2.1 传输方式
 - 4.2.2 协议类型
 - 4.2.3 串行传输格式
 - 4.2.4 数据包格式
 - 4.3 应用层
 - 4.3.1 地址码
 - 4.3.2 功能码
 - 4.3.3 数据域
 - 4.3.4 校验码
 - 4.3.5 应用层功能详解
 - 4.3.5.1 读数据寄存器 (03H)
 - 4.3.5.2 写数据寄存器 (06H)
 - 5 通讯数据表
- 附录 A CRC-16 生成式原理
- 附录 B 通讯应用实例

前言

本使用手册由浙江正泰电器股份有限公司配电电器制造一部提出。

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。本次版本为 V2.2。

NXZ(H)M、NXZB自动转换开关控制器通讯协议使用手册

Modbus-RTU

1 范围

本使用手册规定了 Modbus-RTU 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 NXZ(H)M、NXZB自动转换开关电器产品。

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。

Modbus 协议具有 ASCII 和 RTU 两种传输方式, 我公司生产的断路器采用 RTU 方式。

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	RS485	半双工
通讯地址	1~247 可选	默认: 3
通讯波特率	9.6kbps\19.2kbps\38.4kbps 可选	默认: 9.6kbps
通讯距离	≤ 1000 m	低波特率时
通讯介质	屏蔽双绞线	A 类

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, 偶验位, 1 停止位。**(1 帧数据)

起始	数据								校验	停止
Start	1	2	3	4	5	6	7	8	CRC	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 1~247），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、定义和行为见表2。

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。

CRC16 的生成方法见附录A 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型控制器采集与记录的数据及系统参数。

例如：针对NXZ(H)M型系列，读取常用电源 A、B、C 三相电压值，根据返回数据其结果为 Uan=0001，Ubn=0002，Ucn=0003。

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	
	06	读取寄存器地址低字节		00	数据 1 高字节	
	00	读取数据个数高字节		01	数据 1 低字节	
	03	读取数据个数低字节		00	数据 2 高字节	
校验码	E4	CRC 校验低字节		02	数据 2 低字节	
	28	CRC 校验高字节		00	数据 3 高字节	
				03	数据 3 低字节	
				校验码	E4	CRC 校验低字节
					14	CRC 校验高字节

主机发送 [03 03 00 06 00 03 E4 28]

从机响应 [03 03 06 00 01 00 02 00 03 E4 14]

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：针对NXZ(H)M型系列，将常用电源欠压整定值（寄存器地址为 0x2065）修改为 160V（十六进制为 0x00A0）。

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
	65	待写入地址低字节		65	写入地址低字节
	00	写入数据高字节		00	写入数据高字节
	A0	写入数据低字节		A0	写入数据低字节
校验码	93	CRC 校验低字节	校验码	93	CRC 校验低字节
	8F	CRC 校验高字节		8F	CRC 校验高字节

主机发送 [03 06 20 65 00 A0 93 8F]

从机响应 [03 06 20 65 00 A0 93 8F]

5 通讯数据表

序号	参数项	数据类型	单位	访问规则	地址	参数说明	备注
1	NL1 相电压	UINT	0.1V	R	0x0006	常用电源 NA	NXZ(H)M
2	NL2 相电压	UINT	0.1V	R	0x0007	常用电源 NB	NXZ(H)M
3	NL3 相电压	UINT	0.1V	R	0x0008	常用电源 NC	NXZ(H)M
4	RL1 相电压	UINT	0.1V	R	0x0009	备用电源 RA	NXZ(H)M
5	RL2 相电压	UINT	0.1V	R	0x000A	备用电源 RB	NXZ(H)M
6	RL3 相电压	UINT	0.1V	R	0x000B	备用电源 RC	NXZ(H)M
7	保留项	UINT	\	R	0x000C	返回 0	NXZ(H)M
8	频率	UINT	Hz	R	0x000D	电源频率	NXZ(H)M
9	保留项	UINT	\	R	0x000E	返回 0	NXZ(H)M
10	MAX-N-A 相电压	UINT	1V	R	0x000F	常用 A 最大电压	NXZ(H)M保留
11	MAX-N-B 相电压	UINT	1V	R	0x0010	常用 B 最大电压	NXZ(H)M保留
12	MAX-N-C 相电压	UINT	1V	R	0x0011	常用 C 最大电压	NXZ(H)M保留
13	MAX-R-A 相电压	UINT	1V	R	0x0012	备用 A 最大电压	NXZ(H)M保留
14	MAX-R-B 相电压	UINT	1V	R	0x0013	备用 B 最大电压	NXZ(H)M保留
15	MAX-R-C 相电压	UINT	1V	R	0x0014	备用 C 最大电压	NXZ(H)M保留
16	常用测开关切换次数	UINT	\	R	0x0015	常用开关切换次数	保留
17	备用测开关切换次数	UINT	\	R	0x0016	备用开关切换次数	保留
18	总运行时间	UINT	h	R	0x0017	运行时间	保留
19	ModBus 地址	UINT	NA	R/W	0x0100	地址范围：1~247	NXZ(H)M, NXZB
20	ModBus 波特率	UINT	bps	R/W	0x0101	波特率：0:9.6k; 1:19.2k; 2: 38.4k	NXZ(H)M, NXZB
21	常用、备用电源状态	UINT	NA	R	0x004F	电源状态 ¹	NXZ(H)M, NXZB
22	开关状态	UINT	NA	R	0x0050	开关状态 ²	NXZ(H)M, NXZB
23	常用欠电压 动作阈值整定值 U1	UINT	0.1V	R/W	0x2065	整定范围：160~200	NXZ(H)M
24	备用欠电压 动作阈值整定值 U2	UINT	0.1V	R/W	0x2066	整定范围：160~200	NXZ(H)M
25	常用过电压 动作阈值整定值 U3	UINT	0.1V	R/W	0x2067	整定范围：240~290	NXZ(H)M
26	备用过电压 动作阈值整定值 U4	UINT	0.1V	R/W	0x2068	整定范围：240~290	NXZ(H)M
27	转换延时时间 整定值 T1	UINT	1s	R/W	0x2069	NXZ(H)M: 0~180, NXZB: (0~30s)只读	NXZ(H)M, NXZB
28	返回延时时间 整定值 T2	UINT	1s	R/W	0x206A	NXZ(H)M: 0~180, NXZB: (0~30s)只读	NXZ(H)M, NXZB
29	发电机启动延时 T3	UINT	1s	R/W	0x206B	整定范围：0~180	NXZ(H)M
30	发电机关闭延时 T4	UINT	1s	R/W	0x206C	整定范围：0~180	NXZ(H)M
31	模式选择	UINT	NA	R/W	0x206D	0:电网-电网自投自复 1:电网-电网自投不自 2:电网-发电机自投自复	NXZ(H)M
32	强制转换命令	UINT	NA	W	0x2700	强制转换功能 ⁴	NXZ(H)M, NXZB
33	控制命令	UINT	NA	W	0x2800	控制功能 ³	NXZ(H)M, NXZB

表 A.1 NXZ(H)M 电源状态							
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
保留	保留	常用电源 C 相 00: 正常, 01: 欠压 10: 过压		常用电源 B 相 00: 正常, 01: 欠压 10: 过压		常用电源 A 相 00: 正常, 01: 欠压 10: 过压	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
保留	保留	备用电源 C 相 00: 正常, 01: 欠压 10: 过压		备用电源 B 相 00: 正常, 01: 欠压 10: 过压		备用电源 A 相 00: 正常, 01: 欠压 10: 过压	
续表 A.1 NXZB 电源状态							
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
保留	保留	保留	保留	保留	保留	常用电源 00: 正常, 10: 异常	
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
保留	保留	保留	保留	保留	保留	备用电源 00: 正常, 10: 异常	
表 A.2 开关状态							
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
保留	保留	保留	保留	保留	保留	保留	是否远程控制 0: 否 1: 是
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
故障类型 000: 无 001: 消防联动 010: 电机超时 011: 常用跳闸 100: 备用跳闸			常用侧开 关位置 0: 打开 1: 闭合	备用侧开 关位置 0: 打开 1: 闭合	中间位置 (双分位 置) 0: 否 1: 是	发电机状 态 0: 发电 机组关闭 1: 发电 机组启动	保留
表 A.3 控制命令							
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8
保留	保留	保留	保留	保留	保留	保留	保留
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
保留	保留	清电机超 时故障 0: 否 1: 是	清消防故 障 0: 否 1: 是	保留	远程控制 0: 无 1: 远程 控制	参数设置 0: 无 1: 恢复默 认参数	历史记录设 置 0: 无 1: 清除历史 记录
表 A.4 强制转换命令							
0x0000	强制转换至常用位置		在“远程控制状态”时有 效		说明: NXZ(H)M, NXZB在强制转换时, 即 需要切换到某一路电源合闸时, 必须保 证该路电源电压为正常状态, 产品才能 执行相应动作。		
0x00aa	强制转换至备用位置						
0x00ff	强制转换至双分位置						

附录 A

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，知道执行完了 8 次移位操作，当最后一位 (第 8 位) 移完以后，下一个 8 位字节与寄存器中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：

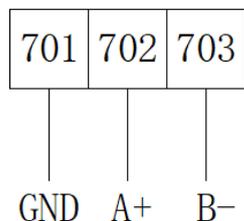
- a) 预置一个 16 位寄存器为 0FFFFH (全 1)，称之为 CRC 寄存器；
- b) 把数据帧中的第一个 8 位字节与 CRC 寄存器中的低字节进行异或运算，结果存回 CRC 寄存器；
- c) 将 CRC 寄存器向右移一位，最高位填以 0，最低位移出并检测；
- d) 如果最低位为 0：重复第三步 (下一次移位)；
- e) 如果最低位为 1：将 CRC 寄存器与一个预设的固定值 (0A001H) 进行异或运算；
- f) 重复第三步和第四步直到 8 次移位。这样处理完了一个完整的八位；
- g) 重复第 2 步到第 5 步来处理下一个八位，直到所有的字节处理结束；
- h) 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

附录 B

NXZ (H) M通讯应用实例——实现远程控制转换至双分位置

B.1 正确的安装调试步骤

(1) 电源与通讯的接线：首先，需将控制器正常上电，保障常备用电压均正常。然后将 RS485 总线中的 A、B 及 GND 线分别和 NXZ(H)M自动转换开关控制器的 702 (A+)、703 (B-)、701 (G) 端子可靠连接；



(2) 按操作说明书，将智能控制器里通讯地址参数调整为 3，波特率调整为 9.6kbps（默认设置通讯地址：3，波特率：9600）；

(3) 将 RS485 总线转换器连接到调试电脑；

(4) 打开串口调试工具，根据实际情况设置串口号、波特率，并将串口参数配置为：1 起始位、8 数据位、偶校验和 1 停止位；

(5) 发送测试帧 (03 06 28 00 00 04 80 4B)，控制器如果返回 (03 06 28 00 00 04 80 4B) 数据，并且控制器屏幕“手动、自动”出现同时闪烁，则说明通讯正常，进入“远程控制状态”。

(6) 发送测试帧 (03 06 27 00 00 FF C2 DC)，控制器返回 (03 06 27 00 00 FF C2 DC)，同时产品强制转换置双分位置（中间位置）。

B.2 通讯异常排查事项

(1) 检查 RS485 通讯总线 A、B 和双电源开关控制器接线端子的 702 (A+)、703 (B-) 是否松开或接反；

(2) 检查双电源开关智能控制器的通讯参数设置是否正确；（应该和上位机一致）；

(3) 检查通讯主机（调试器）里的串口参数设置是否正确；（应该和控制器参数一致）；

(4) 检查通讯适配器（RS485 转换器）是否损坏；（换个新的测试下）；

(5) 如果上述情况都无问题，可与我公司联系进行进一步分析。