# NXZ(H)M and NXZB Automatic Transfer Switch Controller User Manual of the Communication Protocol

**Product name:** Automatic Transfer Switching Equipment

**Model:** NXZM、NXZHM、NXZB、NXZHB series

**Version:** V1.0

**Date:** January 2018

**ZHEJIANG CHINT ELECTRICS CO., LTD.**

## Contents

## Foreword

This user manual is established by Distributing Apparatus Manufacturing Division I of ZHEJIANG CHINT ELECTRICS CO., LTD.

This user manual only represents the information included in this version. No notice will be given for any update. Please pay attention to the latest version of our company. This is version V2.2 of the user manual.

**Communication Protocol of NXZ(H)M and NXZB Automatic Transfer Switch Controller**

**User Manual**

**Modbus-RTU**

## 1 Scope

This user manual specifies the basic terminology, protocol composition and communication datasheet of Modbus-RTU protocol. This user manual is applicable to NXZ(H) M and NXZB Automatic Transfer Switching Equipment.

## 2 Terms and definitions

The following terms are applicable to this user manual.

### 2.1 Open system interconnection (OSI) model

The standard formulated by ISO (International Organization for Standardization) in 1984, with an aim to provide a common basis and standard frame for the interconnection of the computers of different manufacturers .

### 2.2 Physical layer

The first layer of OSI model, which provides a physical link to realize transparent transmission in the communication.

### 2.3 Data-link layer

The second layer in OSI model, which provides transparent and reliable information transmission service between adjacent node.s

### 2.4 Application layer

The seventh layer of OSI model, which realizes the function of data manipulation and information exchange .

### 2.5 Frame

The special information structure composed of several bits or fields pre-established according to a certain standard in data and digital communication. Data are transferred in the network in frame which is a very small unit. Frame is composed of several parts. Different parts have different functions.

## 3 Introduction to Modbus Protocol

Modbus protocol is an industrial bus protocol designed based on ISO/OSI model (7 layers), but only 3 (physical layer, data-link layer and application layer) of the 7 layers are used, which simplifies the protocol model and reduces the difficulty in use.

Modbus protocol has two transmission modes, namely ASCII and RTU. The circuit breaker produced by our company adopts RTU transmission.

## 4 Protocol Overview

### 4.1 Physical layer

| Physical layer parameter | Physical layer content | Remarks |
|---|---|---|
| Communication mode | RS485 | Half duplex |
| Communication address | 1~247 optional | Default: 3 |
| Communication baud rate | 9.6kbps\19.2kbps\38.4kbps optional | Default: 9.6kbps |
| Communication distance | ≤ 1000 m | With a low baud rate |
| Communication media | Shielded Twisted Pair | Category A |

### 4.2 Data-link layer

4.2.1 Transmission mode: master-slave half-duplex mode (the master inquires; the slave responds)

4.2.2 Protocol type: Modbus-RTU communication protocol

4.2.3 Serial transmission format: 1 start bit, 8 data bits, even parity check bit, 1 stop bit (1 frame data)

| Start | Data | | | | | | | | Check | Stop |
|---|---|---|---|---|---|---|---|---|---|---|
| Start | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | CRC | Stop |

4.2.4 Packet (multi-frame) format: as shown in the table

| Start | Address frame | Function frame | Data frame | Check frame | Termination |
|---|---|---|---|---|---|
| T3.5 | 8 bits | 8 bits | n×8 bits | 16 bits | T3.5 |

Note: In RTU model, beginning of information needs at least 3.5-character (or frame) resting time. The resting time can be easily figured out according to the baud rate used (such as T3.5 in the table above). This time delay needs to be considered by UART when it uses Single Chip Micyoco to make the communication protocol. It need not be considered if configuration software or DCS is used, because the bottom layer of the software has already been established.

**4.3 Application layer**

The application layer can make analysis of the packet (including the address code, function code, data field, check code etc.) for the purpose of data exchange.

When the packet sent by the master arrives at theslave, it can enter the addressable device through the communication port. The slave can remove the "envelop" (data head) of the packet and read the valid data. If the data are correct, it will execute the task requested by the data, and add the data it generates to the "envelop" it obtains to form a new packet and send it back to the master. The response data returned include the following: Slave address, executed function, request data generated in command execution, and a check code.

**4.3.1 Address code**

The address code is at the beginning of frame and is composed of 8 bits (take 1~247). These bits mark the address of the slave specified by the user. The slave will receive the data from the master which is connected with it. In the same network, the address of each slave shall be unique, and only the addressable slave will respond to the inquiry including the address. When the slave sends back a response, the address data of the slave that gives the response can tell the master that it's communicating with the master.

**4.3.2 Function code**

The function code can tell which function the addressable slave is executing. All function codes and their definition and behavior are shown in table 2.

| Function code | Definition | Functional behavior |
|---|---|---|
| 03H | Read the data register | Read the data value of one or more registers |
| 06H | Write one register | Write data in one register |

**4.3.3 Data field**

Data field includes the data required by the slave for executing the specific function or the data acquired by the slave when it responds to an inquiry. These data may include values, reference addresses or limiting values. Details can be found in the table of communication address register.

For example: The functional domain tells the slave to read a register. The data field needs to specify the beginning register and read several data. For different types and different capabilities of slaves, embedded addresses and data are also different.

**4.3.4 Check code**

This field allows the master and slave to check the errors during transmission. Sometimes, owing to the electrical noise and other interference, when a set of data are transmitted from one device to the other, they may have some changes. Error check can ensure that the master or slave won't respond to the data that are changed during transmission, which can improve the system safety and efficiency. Error check adopts the 16-bit cyclic redundancy check method.

Cyclic redundancy check (CRC) field occupies two bytes and contains a 16-bit binary value. CRC value is calculated by the transfer equipment and then affixed to the data frame. The receiving equipment calculates CRC value again when the data are received, and then compares it with the value received in CRC field. If the two values are different, it shows there are errors.

Generation method of CRC16 is provided in appendix A CRC-16 generation principle.

### 4.3.5 Explanation of the functions of the application layer

#### 4.3.5.1 Read data register (03H)

03H function code allows the user to acquire the data collected and recorded by the intelligent controller and the system parameters.

For example: for NXZ(H) M series, it can read phases A, B and C voltage value of the normal supply, and according to the data returned, the result is Uan = 0001, Ubn = 0002, Ucn = 0003.

| **Inquiry by the master** | | | **Response by the slave** | | |
|---|---|---|---|---|---|
| **Frame field** | **Content** | **Description** | **Frame field** | **Content** | **Description** |
| Address code | 03 | slave address | Address code | 03 | slave address |
| Function code | 03 | Function code | Function code | 03 | Function code |
| Data field | 00 | Read the high byte of register address | Data field | 06 | Total number of data returned |
| | 06 | Read the low byte of register address | | 00 | High byte of data 1 |
| | 00 | Read the high byte of data number | | 01 | Low byte of data 1 |
| | 03 | Read the low byte of data number | | 00 | High byte of data 2 |
| Check code | E4 | Low byte of CRC check | | 02 | Low byte of data 2 |
| | 28 | High byte of CRC check | | 00 | High byte of data 3 |
| | | | | 03 | Low byte of data 3 |
| | | | Check code | E4 | Low byte of CRC check |
| | | | | 14 | High byte of CRC check |

The master sends [03 03 00 06 00 03 E4 28]

The slave responds to [03 03 06 00 01 00 02 00 03 E4 14]

#### 4.3.5.2 Write data register (06H)

Function code 06H allows the user to modify the content of a single register. Any writable register in an intelligent controller can use this function code to change its value.

For example: For NXZ(H) M series, it can revise the undervoltage setting value of the normal supply (register address: 0x2065) to 160V (hexadecimal number 0x00A0).

| **Inquiry by the master** | | | **Response by the slave** | | |
|---|---|---|---|---|---|
| **Frame field** | **Content** | **Description** | **Frame field** | **Content** | **Description** |
| Address code | 03 | Slave address | Address code | 03 | Slave address |
| Function code | 06 | Function code | Function code | 06 | Function code |
| Data field | 20 | High byte of the address to be written in | Data field | 20 | High byte of the address written in |
| | 65 | Low byte of the address to be written in | | 65 | Low byte of the address written in |
| | 00 | High byte of read-in data | | 00 | High byte of read-in data |
| | A0 | Low byte of read-in data | | A0 | Low byte of read-in data |
| Check code | 93 | Low byte of CRC check | Check code | 93 | Low byte of CRC check |
| | 8F | High byte of CRC check | | 8F | High byte of CRC check |

The master sends [03 06 20 65 00 A0 93 8F].

The slave responds to [03 06 20 65 00 A0 93 8F].

**5 Communication Datasheet**

| No. | Parameter | Data type | Unit | Access rule | Address | Parameter specification | Remarks |
|---|---|---|---|---|---|---|---|
| 1 | NL1 phase voltage | UINT | 0.1V | R | 0x0006 | Normal supply NA | NXZ(H)M |
| 2 | NL2 phase voltage | UINT | 0.1V | R | 0x0007 | Normal supply NB | NXZ(H)M |
| 3 | NL3 phase voltage | UINT | 0.1V | R | 0x0008 | Normal supply NC | NXZ(H)M |
| 4 | RL1 phase voltage | UINT | 0.1V | R | 0x0009 | Alternative supply RA | NXZ(H)M |
| 5 | RL2 phase voltage | UINT | 0.1V | R | 0x000A | Alternative supply RB | NXZ(H)M |
| 6 | RL3 phase voltage | UINT | 0.1V | R | 0x000B | Alternative supply RC | NXZ(H)M |
| 7 | Reserved item | UINT | \ | R | 0x000C | Return to 0 | NXZ(H)M |
| 8 | Frequency | UINT | Hz | R | 0x000D | Power frequency | NXZ(H)M |
| 9 | Reserved item | UINT | \ | R | Ox000E | Return to 0 | NXZ(H)M |
| 10 | MAX-N-A phase voltage | UINT | 1V | R | 0x000F | Common A peak voltage | NXZ(H)M reserved |
| 11 | MAX-N-B phase voltage | UINT | 1V | R | 0x0010 | Common B peak voltage | NXZ(H)M reserved |
| 12 | MAX-N-C phase voltage | UINT | 1V | R | 0x0011 | Common C peak voltage | NXZ(H)M reserved |
| 13 | MAX-R-A phase voltage | UINT | 1V | R | 0x0012 | Standby A peak voltage | NXZ(H)M reserved |
| 14 | MAX-R-B phase voltage | UINT | 1V | R | 0x0013 | Standby B peak voltage | NXZ(H)M reserved |
| 15 | MAX-R-C phase voltage | UINT | 1V | R | 0x0014 | Standby C peak voltage | NXZ(H)M reserved |
| 16 | Switching times on the common side | UINT | \ | R | 0x0015 | Switching times of the common switch | Reserved |
| 17 | Switching times on the standby side | UINT | \ | R | 0x0016 | Switching times of the standby switch | Reserved |
| 18 | Total operation time | UINT | h | R | 0x0017 | Operation time | Reserved |
| 19 | ModBus address | UINT | NA | R/W | 0x0100 | Address range: 1～247 | NXZ(H)M,NXZB |
| 20 | ModBus baud rate | UINT | bps | R/W | 0x0101 | Baud rate: 0:9.6k; 1:19.2k; 2: 38.4k | NXZ(H)M, NXZB |
| 21 | Normal/alternative supply state | UINT | NA | R | 0x004F | Power supply state [1] | NXZ(H)M,NXZB |
| 22 | Switching state | UINT | NA | R | 0x0050 | Switching state [2] | NXZ(H)M,NXZB |
| 23 | Setting value of common under-voltage threshold U1 | UINT | 0.1V | R/W | 0x2065 | Setting range: 160～200 | NXZ(H)M |
| 24 | Setting value of standby undervoltage threshold U2 | UINT | 0.1V | R/W | 0x2066 | Setting range: 160～200 | NXZ(H)M |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 25 | Setting value of common overvoltage threshold U3 | UINT | 0.1V | R/W | 0x2067 | Setting range: 240～290 | NXZ(H)M |
| 26 | Setting value of standby overvoltage threshold U4 | UINT | 0.1V | R/W | 0x2068 | Setting range: 240～290 | NXZ(H)M |
| 27 | Transfer delay time setting T1 | UINT | 1s | R/W | 0x2069 | NXZ(H)M: 0～180, NXZB: (0~30s) Read Only | NXZ(H)M,NXZB |
| 28 | Return delay time setting T2 | UINT | 1s | R/W | 0x206A | NXZ(H)M: 0～180, NXZB: (0~30s)Read Only | NXZ(H)M,NXZB |
| 29 | Generator startup delay T3 | UINT | 1s | R/W | 0x206B | Setting range: 0～180 | NXZ(H)M |
| 30 | Generator shutdown delay T4 | UINT | 1s | R/W | 0x206C | Setting range: 0～180 | NXZ(H)M |
| 31 | Mode selection | UINT | NA | R/W | 0x206D | 0: Power grid-power grid automatic switching automatic recovery<br>1: Power grid-power grid automatic switching nonautomatic recovery<br>2: Power grid-generator automatic switching automatic recovery | NXZ(H)M |
| 32 | Compulsory switch command | UINT | NA | W | 0x2700 | Compulsory switch function [4] | NXZ(H)M,NXZB |
| 33 | Control command | UINT | NA | W | 0x2800 | Control function [3] | NXZ(H)M,NXZB |

**Table A.1 NXZ(H) M Power Supply State**

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
|---|---|---|---|---|---|---|---|
| Reserved | Reserved | Normal supply C phase 00: normal, 01: undervoltage, 10: overvoltage | | Normal supply B phase 00: normal, 01: undervoltage, 10: overvoltage | | Normal supply A phase 00: normal, 01: undervoltage, 10: overvoltage | |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| Reserved | Reserved | Alternative supply C phase 00: normal, 01: undervoltage, 10: | | Alternative supply B phase 00: normal, 01: undervoltage, 10: overvoltage | | Alternative supply A phase 00: normal, 01: undervoltage, 10: | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | overvoltage | | | | overvoltage | |
| **Continued Table A.1 NXZB Power Supply State** | | | | | | | |
| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Normal supply<br>00: normal, 10: abnormal | |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Alternative supply<br>00: normal, 10: abnormal | |
| **Table A.2 Switching State** | | | | | | | |
| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Remote control or not<br>0: no 1: yes |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| Fault type<br>000: NA<br>001: Fire control linkage<br>010: Motor timeout<br>011: Common trip<br>100: Standby trip | | | Switch position on the common side<br>0: open 1: closed | Switch position on the standby side<br>0: open 1: closed | Middle position (double tripping position）<br>0: no 1: yes | Generator state<br>0: genset shut down<br>1: genset started | Reserved |
| **Table A.3 Control Command** | | | | | | | |
| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
| Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved | Reserved |
| Bit7 | Bit6 | Bit5 | Bit4 | Bit3 | Bit2 | Bit1 | Bit0 |
| Reserved | Reserved | Clearance of motor timeout fault<br>0: no 1: yes | Clearance of fire control fault<br>0: no 1: yes | Reserved | Remote control<br>0: NA 1: remote control | Parameter setting<br>0: NA 1: restored to default parameter | Historic record setting<br>0: NA 1: clearance of historic records |
| **Table A.4 Compulsory Switch Command** | | | | | | | |
| 0x0000 | Compulsory switch to common position | | Valid under the "remote control state" | | Note: During compulsory switch of NXZ(H) M, NXZB, namely, when switching to the switching-on position, the voltage of the power supply shall be in a normal condition; in this way, the product can execute relevant action. | | |
| 0x00aa | Compulsory switch to standby position | | | | | | |
| 0x00ff | Compulsory switch to double tripping position | | | | | | |

# Appendix A
## CRC-16 Generation Principle

During CRC calculation, firstly, it pre-sets a 16-bit register to 1 totally, and then it continuously calculates the 8-bit byte in the data frame and the current value in the register. Only 8 data bits of each byte are involved to generate CRC. The start bit, stop bit, and the parity bit possibly used will not affect CRC.

During generation of CRC, each 8-bit byte executes XOR with the content in the register and then shifts the result to a lower bit. The high bit is supplemented with "0". The least significant bit (LSB) is shifted out and checked. If the LSB is 1, the register will have an XOR operation with a preset fixed value. If LSB is 0, no actions will be taken.

The above procedures will be repeated, till 8 shift operations are finished. When the last bit (the 8th bit) is shifted, the next 8-bit byte will have an XOR operation with the current value of the register. Another 8 shift XOR operations will be repeated. When all bytes in the data frame are handled, the end value generated is CRC value.

The process of generation of a CRC value is:

a) Pre-set a 16-bit register as 0FFFFH (total 1), which is called CRC register;

b) Execute XOR operation between the first 8-bit byte in the data frame with the low byte in the CRC register, and store the result in CRC register;

c) Shift CRC register rightward for one bit, fill 0 to the most significant bit, and shift the least significant bit out and check it;

d) If the least significant bit is 0, repeat step 3 (make a shift next time);

e) If the least significant bit is 1, execute XOR operation between CRC register and a preset fixed value (0A001H);

f) Repeat step 3 and step 4, till the 8th shift. In this way, operation of a complete 8 bits will be finished;

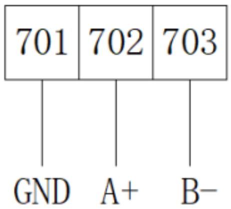
g) Repeat steps 2~5 to deal with the next 8-bit byte, till all bytes are processed;

h) The final CRC register value is then the CRC value.

## Appendix B
## Example of Application of NXZ(H) M Communication---Switch of Remote Control to Double Tripping Position

### B.1 Correct Installation and Debugging Procedure

(1) Wiring between the power supply and communication: First of all, power on the controller normally to ensure the common and standby voltages are normal. Then reliably connect wires A, B and GND in RS485 bus with 702 (A+), 703 (B-) and 701 (G) terminal of NXZ(H) M automatic transfer switch controller.




(2) Adjust the communication address parameter in the intelligent controller to 3 and adjust the baud rate to 9.6kbps (the default communication address parameter is 3 and baud rate is 9600) according to the user manual.

(3) Connect RS485 bus converter to the debugging computer.

(4) Start the serial port debugging tool, and set the serial port No. and baud rate according to actual conditions. Configure the serial port parameter as: 1 start bit, 8 data bits, even parity check and 1 stop bit.

(5) Send the testing frame（03 06 28 00 00 04 80 4B).If the controller returns (03 06 28 00 00 04 80 4B) data, and the "manual" and "automatic" buttons on the controller screen blink at the same time, it means the communication is normal, and it enters the "remote control state".

(6) Send the testing frame（03 06 27 00 00 FF C2 DC). The controller will return (03 06 27 00 00 FF C2 DC), and at the same time, the product will be compulsorily switched to the double tripping position (middle position).

### B.2 Inspection of Communication Anomaly

(1) Check if the connection between RS485 communication buses A and B and the terminal 702 (A+), 703 (B-) of the dual power switch intelligent controller is loose or reverse.

(2) Check if the communication parameters of the dual power switch intelligent controller are correctly set (they shall be consistent with those of the upper computer).

(3) Check if the serial port parameters of the master (debugger) are correctly set (they shall be consistent with the parameters of the controller).

(4) Check if the communication adapter (RS485 convertor) is damaged (change for a new one to be tested).

(5) If the above problems do not exist, please contact with us for a further analysis.