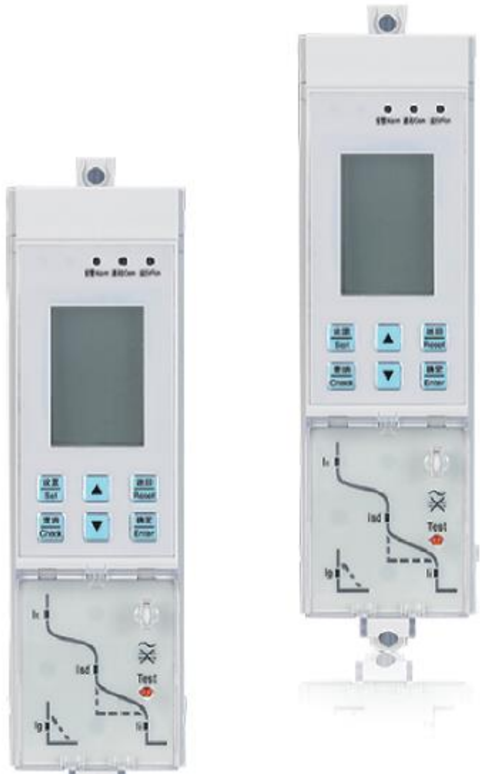
# H Intelligent Controller Communication Protocol

——Modbus-RTU

## User Manual



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

**06-2019**

# Catalog

# Foreword

This document is put forward by the secondary distributing apparatus manufacturing department, ZHEJIANG CHINT ELECTRICS CO.,LTD.

Our company reserves the right to update this document without prior notice. Please check for latest edition.

# H Intelligent Controller Communication Protocol

## ——Modbus-RTU

## 1 Scope

This document specifies the basic terminology, protocol content and communication data tables of Modbus-RTU.

This document applies to NA8 series ACB with 3H controller.

## 2 Definitions

The terminology below is applicable to this document.

### 2.1 open system interconnection（OSI）model

International standardization organization (ISO) set the standard in 1984 to provide a common base and a standard framework for computers from different manufacturers.

### 2.2 physical layer

Layer 1 of OSI model; provides physical link for communication.

### 2.3 data link layer (DDL)

Layer 2 of OSI model; provides transparent and reliable information transmission service between adjacent nodes.

### 2.4 application layer

Layer 7 of OSI model; realizes the function of data manipulation and information exchange.

### 2.5 Frame

In data and digital communication, a frame is a specific information structure composed of several bits or fields according to a standard. Data on the internet is transmitted frame by frame. Different parts of frame have different functions.

## 3 Modbus protocol introduction

Modbus is an industrial bus protocol based on ISO/OSI model (7 layers). However, only 3 layers (physical layer, data link layer and application layer) are used, which helps to simplify protocol model and reduce the difficult to use it.

Modbus has two transmission modes, ASCII and RTU. In this document, we adopt RTU mode.

## 4 protocol introduction

### 4.1 physical layer

| physical layer parameter | **content** | **remark** |
|---|---|---|
| Communication mode | RS-485 | 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 | Under situation of low baud rate |
| Communication medium | Shielded twisted pair | A class |
| Max connections | 32 | Computer networking |

4.2 data link layer

4.2.1 Transmission mode: half duplex protocol (host computer queries and slave computers reply)

4.2.2 Protocol type: Modbus-RTU.

4.2.3 Serial transmission format: 1 start bit, 8 data bits, no check bit and 2 stop bits. (A frame of data)

| **start** | **data** | | | | | | | | **stop** | |
|---|---|---|---|---|---|---|---|---|---|---|
| Start | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Stop | Stop |

4.2.4 Data packet (multiple frames) format:

| **start** | **address** | **function** | **data** | **check** | **stop** |
|---|---|---|---|---|---|
| T3.5 | 8 bits | 8 bits | n×8 bits | 16 bits | T3.5 |

Note: in RTU mode, information starts with 3.5-characters (frames) quiescent time at least. The quiescent time, such as T3.5 shown in table above, can be calculated easily according to the used baud rate. The delay should be taken into consideration, if using UART to establish a communication protocol. It can be ignored if configuration software or DCS is adopted, because underlying part is accomplished.

4.3 application layer

Application layer can parse the content of packet to exchange data.

When the slave computer receives the packet from the host computer, the packet will enter addressing equipment by communication port. The slave computer will strip the data headers to read valid data. If there is no error, the salve computer will execute the task requested by data, add the new-generated data into "envelop" to form a new data packet and send the packet to the host computer. The data returned includes the salve computer address (Address), executed function (Function), requested data generated by executing command (Data) and a check code (Check).

4.3.1 Address code

Address code, in the beginning of frame, is an 8-bit code (1-247). The code addresses the user specified slave computer and the salve computer can receive data from the connected host computer. The address of every salve computer must be exclusive. The exclusively addressed terminal will respond to the inquiry including this address. When a response is sent back, the subordinate address data included in the response will assure the host computer which device is communicating with it.

4.3.2 Function code

The function code tells the located terminal of the function to be executed. All function codes, their definitions and initial functions are listed below.

| Code | Definition | Task |
|---|---|---|
| 03H | Reading data register | Obtain the binary value of one or more registers |
| 06H | Preset single register | Put a certain given binary value into a register |

4.3.3 Data Domain

The data domain contains the data needed by the terminal to execute a certain function or the data gathered by the terminal in responding an inquiry. These data could be numerical value, reference address or extreme value. For example, the function domain code tells the terminal to read a register, while the data domain indicates which register to start and how many data to read. The inline address and data varies according to types and the subordinate devices.

4.3.4 Check code

This domain makes possible the principle and terminal to check errors during transmission. Because of electrical noise or other jams, some changes may happen to data on road from one device to another. The check domain makes sure neither the principle nor the subordinate respond to changed data, which improves the safety and efficiency of the system. The check applies 16-bit-circulation-redundance.

Circulation-Redundancy-Check （CRC） domain occupies two bits, including a 16-digit binary value. The transmission device calculates out the CRC value, adds it to the data frame; the receiving device also calculates the CRC value, compares it with the received CRC domain value, and if these two values don't match, it proves a mistake has occurred.

Note: generation method of CRC16 please refers to Appendix F1 (CRC-16 generation principle)

4.3.5 Application Layer Explanation

4.3.5.1 Read Register（03H）

Any data and system parameters collected and reported by controller is available to user by 03 function code.

For example, read Ia，Ib，Ic and get the result of Ia=0001，Ib=0002，Ic=0003, according to returned data.

| Host computer queries | | | Salve computes responds | | |
|---|---|---|---|---|---|
| Frame domain | content | explanation | Frame domain | content | explanation |
| Address code | 03 | Salve computer address | Address code | 03 | Salve computer address |
| Function Code | 03 | Function Code | Function Code | 03 | Function Code |
| Data Domain | 00 | Read high byte of register address | Data Domain | 06 | Return the total number of data |
| | 01 | Read low byte of register address | | 00 | high byte of data1 |
| | | | | 01 | low byte of data1 |
| | 00 | Read high byte of number of data | | 00 | high byte of data2 |
| | | | | 02 | low byte of data2 |
| | 03 | Read low byte of number of data | | 00 | high byte of data3 |
| | | | | 03 | low byte of data3 |

| | | | | | |
|---|---|---|---|---|---|
| Check code | 55 | CRC checks low byte | Check code | E4 | CRC checks low byte |
| | E9 | CRC checks high byte | | 14 | CRC checks high byte |

Host computer sends [03 03 00 01 00 03 55 e9]

Salve computer responds [03 03 06 00 01 00 02 00 03 e4 14]

4.3.5.2 Preset single register（06H）

Function code 06 makes possible the user change the content of the single register. Every single register in the DAE system can use this command to change values.

For example, change long delay setting current (register address: v0x2007) to 2000A (hex: 0x07d0).

| **Host computer queries** | | | **Salve computes responds** | | |
|---|---|---|---|---|---|
| **Frame domain** | **content** | **explanation** | **Frame domain** | **content** | **explanation** |
| Address code | 03 | Salve computer address | Address code | 03 | Salve computer address |
| Function Code | 06 | Function Code | Function Code | 06 | Function Code |
| Data Domain | 20 | Address high byte to write | Data Domain | 20 | Write high byte of address |
| | 07 | Address low byte to write | | 07 | Write low byte of address |
| | 07 | Write data high byte | | 07 | Write data high byte |
| | D0 | Write data low byte | | D0 | Write data low byte |
| Check code | 31 | CRC checks low byte | Check code | 31 | CRC checks low byte |
| | 85 | CRC checks high byte | | 85 | CRC checks high byte |

Host computer sends [03 06 20 07 07 d0 31 85]

Salve computer responds [03 06 20 07 07 d0 31 85]

# 5 Comm. Address List

WORD is word-size. UINT is unsigned integer. SINT is signed integer. ULONG is long integer. ULONG LONG is double long integer. BCD is BCD code. R is read only. W is write only. R/W is read-write.

## 5.1 Measuring message

| No. | Content | Data type | Unit | Attribute | Address | Remark |
|---|---|---|---|---|---|---|
| 1 | Operating condition word | WORD | - | R | 0x0000 | See Appendix A |
| 2 | L1 current | UINT | A | R | 0x0001 | NOTE 1 |
| 3 | L2 current | UINT | A | R | 0x0002 | NOTE 1 |
| 4 | L3 current | UINT | A | R | 0x0003 | NOTE 1 |
| 5 | LN current | UINT | A | R | 0x0004 | NOTE 1 |
| 6 | LG current | UINT | A | R | 0x0005 | NOTE 1 |
| 7 | L1 voltage | UINT | V | R | 0x0006 | ×0.1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 8 | L2 voltage | UINT | V | R | 0x0007 | ×0.1 |
| 9 | L3 voltage | UINT | V | R | 0x0008 | ×0.1 |
| 10 | L1-2 voltage | UINT | V | R | 0x0009 | ×0.1 |
| 11 | L2-3 voltage | UINT | V | R | 0x000A | ×0.1 |
| 12 | L3-1 voltage | UINT | V | R | 0x000B | ×0.1 |
| 13 | Power factor | SINT | % | R | 0x000C | ×0.01 |
| 14 | frequency | UINT | Hz | R | 0x000D | ×0.01 |
| 15 | | | | | | |
| 16 | | | | | | |
| 17 | Line voltage average | UINT | V | R | 0x0010 | ×0.1 |
| 18 | Unbalanced rate of current | UINT | % | R | 0x0011 | ×0.1 |
| 19 | | | | | | |
| 20 | Phase L1 active power | SINT | kW | R | 0x0021 | NOTE 1 |
| 21 | Phase L1 reactive power | SINT | kVar | R | 0x0022 | NOTE 1 |
| 22 | Phase L1 apparent power | UINT | KVA | R | 0x0023 | NOTE 1 |
| 23 | Phase L2 active power | SINT | kW | R | 0x0024 | NOTE 1 |
| 24 | Phase L2 reactive power | SINT | kVar | R | 0x0025 | NOTE 1 |
| 25 | Phase L2 apparent power | UINT | kVA | R | 0x0026 | NOTE 1 |
| 26 | Phase L3 active power | SINT | kW | R | 0x0027 | NOTE 1 |
| 27 | Phase L3 reactive power | SINT | kVar | R | 0x0028 | NOTE 1 |
| 28 | Phase L3 apparent power | UINT | kVA | R | 0x0029 | NOTE 1 |
| 29 | The total active power | SINT | kW | R | 0x002A | NOTE 1 |
| 30 | The total reactive power | SINT | kVar | R | 0x002B | NOTE 1 |
| 31 | The total apparent power | UINT | KVA | R | 0x002C | NOTE 1 |
| 32 | | | | | | |
| 33 | | | | | | |
| 34 | | | | | | |

NOTE 1: when frame current is 2000, value×1 . With other frame current, value×2 .

Appendix A Bit format of operating conditions word

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | - | - |
| **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| - | - | Open/close indicating status bit | Trip indicating status bit | - | - | - | - |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | 0：close<br>1：open | 0：normal<br>1：fault | | | | |

5.2 System Maintenance

| No. | Content | Data type | Unit | Attribute | Address | Remark |
|---|---|---|---|---|---|---|
| 1 | ModBus address | UINT | - | R/W | 0x0100 | |
| 2 | ModBus baud rate | UINT | - | R/W | 0x0101 | |
| 3 | | | | | | |
| 4 | System time(year\month) | BCD\BCD | - | W | 0x0140 | High 8 bit\low 8 bit |
| 5 | System time (date\hour) | BCD\BCD | - | W | 0x0141 | |
| 6 | System time (min\sec) | BCD\BCD | - | W | 0x0142 | |
| 7 | | | - | | | |
| 8 | Frame current | UINT | A | R | 0x0180 | |
| 9 | Rated current | UINT | A | R | 0x0181 | NOTE 3 |
| 10 | Rated voltage | UINT | V | R | 0x0182 | |
| 11 | | | | | | |
| 12 | Description ID | BCD | - | R | 0x0186 ～ 0x0188 | See Appendix G |

Note 3：when frame current is 2000, value×1 . With other frame current, value×2 .

5.3 fault record

| No. | Content | Data type | Unit | Attribute | Address | Remark |
|---|---|---|---|---|---|---|
| 1 | 1st trip time(year\month) | BCD\BCD | - | R | 0x0200 | |
| 2 | 1st trip time (date\hour) | BCD\BCD | - | R | 0x0201 | |
| 3 | 1st trip time (min\sec) | BCD\BCD | - | R | 0x0202 | |
| 4 | Reason for 1st trip | WORD | - | R | 0x0203 | See Appendix B |
| 5 | | | | | | |
| 6 | L1 current of 1st trip | UINT | A | R | 0x0208 | Note 4 |
| 7 | L2 current of 1st trip | UINT | A | R | 0x0209 | Note 4 |
| 8 | L3 current of 1st trip | UINT | A | R | 0x0210 | Note 4 |
| 9 | L4 current of 1st trip | UINT | A | R | 0x0211 | Note 4 |
| 10 | | | | | | |
| 11 | 1st trip time | UINT | s | R | 0x0213 | ×0.01 |
| 12 | | | | | | |
| 13 | 2nd trip time(year\month) | BCD\BCD | | R | 0x0214 | |
| 14 | 2nd trip time (date\hour) | BCD\BCD | | R | 0x0215 | |
| 15 | 2nd trip time (min\sec) | BCD\BCD | | R | 0x0216 | |
| 16 | Reason for 2nd trip | WORD | | R | 0x0217 | See Appendix B |
| 17 | | | | | | |
| 18 | L1 current of 2nd trip | UINT | A | R | 0x021C | Note 4 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 19 | L2 current of 2nd trip | UINT | A | R | 0x021D | Note 4 |
| 20 | L3 current of 2nd trip | UINT | A | R | 0x021E | Note 4 |
| 21 | L4 current of 2nd trip | UINT | A | R | 0x021F | Note 4 |
| 22 | | | | | | |
| 23 | 2nd trip time | UINT | s | R | 0x0227 | ×0.01 |
| 24 | | | | | | |
| 25 | 3rd trip time(year\month) | BCD\BCD | | R | 0x0228 | |
| 26 | 3rd trip time (date\hour) | BCD\BCD | | R | 0x0229 | |
| 27 | 3rd trip time (min\sec) | BCD\BCD | | R | 0x022A | |
| 28 | Reason for 3rd trip | WORD | | R | 0x022B | See Appendix B |
| 29 | | | | | | |
| 30 | 3rd trip time | UINT | s | R | 0x0230 | ×0.01 |
| 31 | | | | | | |
| 32 | 4th trip time(year\month) | BCD\BCD | | R | 0x0231 | |
| 33 | 4th trip time (date\hour) | BCD\BCD | | R | 0x0232 | |
| 34 | 4th trip time (min\sec) | BCD\BCD | | R | 0x0233 | |
| 35 | Reason for 4th trip | WORD | | R | 0x0234 | See Appendix B |
| 36 | | | | | | |
| 37 | 4th trip time | UINT | s | R | 0x0239 | ×0.01 |
| 38 | | | | | | |
| 39 | 5th trip time(year\month) | BCD\BCD | | R | 0x023A | |
| 40 | 5th trip time (date\hour) | BCD\BCD | | R | 0x023B | |
| 41 | 5th trip time (min\sec) | BCD\BCD | | R | 0x023C | |
| 42 | Reason for 5th trip | WORD | | R | 0x023D | See Appendix B |
| 43 | | | | | | |
| 44 | 5th trip time | UINT | s | R | 0x0242 | ×0.01 |
| 45 | | | | | | |
| 46 | 6th trip time(year\month) | BCD\BCD | | R | 0x0243 | |
| 47 | 6th trip time (date\hour) | BCD\BCD | | R | 0x0244 | |
| 48 | 6th trip time (min\sec) | BCD\BCD | | R | 0x0245 | |
| 49 | Reason for 6th trip | WORD | | R | 0x0246 | See Appendix B |
| 50 | | | | | | |
| 51 | 6th trip time | UINT | s | R | 0x024B | ×0.01 |
| 52 | | | | | | |
| 53 | 7th trip time(year\month) | BCD\BCD | | R | 0x024C | |
| 54 | 7th trip time (date\hour) | BCD\BCD | | R | 0x024D | |
| 55 | 7th trip time (min\sec) | BCD\BCD | | R | 0x024E | |
| 56 | Reason for 7th trip | WORD | | R | 0x024F | See Appendix B |

| | | | | | | |
|---|---|---|---|---|---|---|
| 57 | | | | | | |
| 58 | 7th trip time | UINT | s | R | 0x0254 | ×0.01 |
| 59 | | | | | | |
| 60 | 8th trip time(year\month) | BCD\BCD | | R | 0x0255 | |
| 61 | 8th trip time (date\hour) | BCD\BCD | | R | 0x0256 | |
| 62 | 8th trip time (min\sec) | BCD\BCD | | R | 0x0257 | |
| 63 | Reason for 8th trip | WORD | | R | 0x0258 | See Appendix B |
| 64 | | | | | | |
| 65 | 8th trip time | UINT | s | R | 0x025D | ×0.01 |

NOTE 4： when frame current is 2000, value×1 . With other frame current , value×2 .

Appendix B Fault Type Sheet

| Code | Type | remark |
|---|---|---|
| 00H(hex) | Fault free | 0(dec) |
| 01H | Overload long delay fault | 1 |
| 02H | Short circuit short delay fault | 2 |
| 03H | Short circuit transient fault | 3 |
| 04H | Earth fault | 4 |
| 05H | Leakage trip fault | 5 |
| 06H | Neutral fault | 6 |
| 07H | Unbalanced current fault | 7 |
| 08H | MCR action | 8 |
| 09H | Short circuit interlocking fault | 9 |
| 0AH | Earth interlocking fault | 10 |
| 0BH | Undervoltage fault | 11 |
| 0CH | Overvoltage fault | 12 |
| 0DH | Unbalanced voltage fault | 13 |
| 0EH | Under frequency fault | 14 |
| 0FH | Over frequency fault | 15 |
| 10H | Phase sequence fault | 16 |
| 11H | Reverse power fault | 17 |

5.4 parameter setting

| No. | Content | Data type | Unit | Attribute | Address | Remark |
|---|---|---|---|---|---|---|
| 1 | Protection function set 0 (current basic protection) | | - | R/W | 0x2000 | See Appendix C |
| 2 | Protection function set 1 | | - | R/W | 0x2001 | See Appendix D |
| | Protection function set 2 | | - | R/W | 0x2002 | See Appendix E |
| | | | | | | |
| 3 | Long delay setting | UINT | A | R/W | 0x2007 | Note5-7 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | current | | | | | |
| 4 | Long delay setting time | UINT | s | R/W | 0x2008 | Note 6 |
| 5 | Short delay setting current | UINT | A | R/W | 0x2009 | Note5-7 |
| 6 | Short delay setting time | UINT | ms | R/W | 0x200A | Note 6 |
| 7 | instantaneous current | UINT | A | R/W | 0x200B | Note5-7 |
| 8 | Earth protection setting current | UINT | A | R/W | 0x200C | Note6-7 |
| 9 | Earth protection setting time | UINT | ms | R/W | 0x200D | Note 6 |
| 10 | I unbalance starting value | UINT | % | R/W | 0x200E | |
| 11 | I unbalance starting time | UINT | s | R/W | 0x200F | ×0.02 |
| 12 | I unbalance returned value | UINT | % | R/W | 0x2010 | |
| 13 | I unbalance returned time | UINT | s | R/W | 0x2011 | ×0.02 |
| 14 | | | | | | |
| 15 | U unbalance starting value | UINT | % | R/W | 0x2016 | |
| 16 | U unbalance starting time | UINT | s | R/W | 0x2017 | ×0.02 |
| 17 | U unbalance returned value | UINT | % | R/W | 0x2018 | |
| 18 | U unbalance returned time | UINT | s | R/W | 0x2019 | ×0.02 |
| 19 | | | | | | |
| 20 | Undervoltage starting value | UINT | V | R/W | 0x201A | |
| 21 | Undervoltage starting time | UINT | s | R/W | 0x201B | ×0.02 |
| 22 | Undervoltage returned value | UINT | V | R/W | 0x201C | |
| 23 | Undervoltage returned time | UINT | s | R/W | 0x201D | ×0.02 |
| 24 | | | | | | |
| 25 | Overvoltage starting value | UINT | V | R/W | 0x201E | |
| 26 | Overvoltage starting time | UINT | s | R/W | 0x201F | ×0.02 |
| 27 | Overvoltage returned value | UINT | V | R/W | 0x2020 | |
| 28 | Overvoltage returned time | UINT | s | R/W | 0x2021 | ×0.02 |
| 29 | | | | | | |
| 30 | Under frequency starting value | UINT | Hz | R/W | 0x2022 | ×0.1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 31 | Under frequency starting time | UINT | s | R/W | 0x2023 | ×0.02 |
| 32 | Under frequency returned value | UINT | Hz | R/W | 0x2024 | ×0.1 |
| 33 | Under frequency returned time | UINT | s | R/W | 0x2025 | ×0.02 |
| 34 | | | | | | |
| 35 | Over frequency starting value | UINT | Hz | R/W | 0x2026 | ×0.1 |
| 36 | Over frequency starting time | UINT | s | R/W | 0x2027 | ×0.02 |
| 37 | Over frequency returned value | UINT | Hz | R/W | 0x2028 | ×0.1 |
| 38 | Over frequency returned time | UINT | s | R/W | 0x2029 | ×0.02 |

Note 5：when frame current is 2000, value×1 . With other frame current , value×2 .

Note 6：setting range of each parameter please refer to section 3 of 《NA8 multi-functional intelligent controller》.

Note 7： 65535（0XFFFF）means the function is turned off.

Appendix C Bit format of protection function set

| **Bit15** | **Bit14** | **Bit13** | **Bit12** | **Bit11** | **Bit10** | **Bit9** | **Bit8** |
|---|---|---|---|---|---|---|---|
| Long delay cooling time<br>000：instant<br>001：10min<br>010：20min<br>011：30min | | | - | - | - | Long delay protection curve type<br>00：$I^2T$<br>01：IT<br>10：$I^4T$ | |
| **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| - | - | - | - | - | - | - | - |

Appendix D Protection function set 1

| **Bit15** | **Bit14** | **Bit13** | **Bit12** | **Bit11** | **Bit10** | **Bit9** | **Bit8** |
|---|---|---|---|---|---|---|---|
| | | | | I unbalance protection<br>00：turned off<br>01：trip<br>10：alarm | | | |
| **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| | | | | | | | |

Appendix E Protection function set 2

| **Bit15** | **Bit14** | **Bit13** | **Bit12** | **Bit11** | **Bit10** | **Bit9** | **Bit8** |
|---|---|---|---|---|---|---|---|
| Undervoltage protection<br>00：turned off | | Overvoltage protection<br>00：turned off | | | | | |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01：trip<br>10：alarm | | 01：trip<br>10：alarm | | | | | |
| **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| U unbalance protection<br>00：turned off<br>01：trip<br>10：alarm | | | | Under frequency protection<br>00：turned off<br>01：trip<br>10：alarm | | Over frequency protection<br>00：turned off<br>01：trip<br>10：alarm | |

5.5 remote control

| No. | Content | Data type | Unit | Attribute | Address | Remark |
|---|---|---|---|---|---|---|
| 1 | Control command | WORD | - | W | 0x2800 | See Appendix F |

Appendix F Bit format of control command

| Bit15 | Bit14 | Bit13 | Bit12 | Bit11 | Bit10 | Bit9 | Bit8 |
|---|---|---|---|---|---|---|---|
| - | - | - | - | - | - | Open/close control<br>01：open<br>10：close | |
| **Bit7** | **Bit6** | **Bit5** | **Bit4** | **Bit3** | **Bit2** | **Bit1** | **Bit0** |
| - | - | - | - | - | - | - | - |

Appendix G Description ID

| 0x0188 | | 0x0187 | | 0x0186 | |
|---|---|---|---|---|---|
| H | L | H | L | H | L |
| 20 | 16 | 05 | 13 | 25 | 90 |

Note: factory number shown in the table above is 201605132590.

**Appendix F1 CRC-16 generation principle**

CRC set every digit of a 16-bit register to 1, calculates the 8 digits of the data frame and the current value of the register; only 8 bits of every byte are involved in the make of the CRC. The initial bit, end bit and parity bit are irrelevant to CRC.

In the make of CRC, every 8-bit byte carries out “ XOR ” operation with the content of the register, shift the result to the lower bit, supplement the higher bit with “0”, measure the lowest bit（LSB）, and if 1, this register carries an “ XOR ” operation with a pre-set fixed value, and if the LSB is 0, no treatment is to be given.

Repeat the above circle 8 operations. Next 8-bit byte carries out an “XOR” operation with the current value of the register and do the above circle after the last bit in the data frame are shifted like this. The finally generated value is the CRC after all byte in data frame was done with the above circle operations.

The making process of a CRC:

a) Preset a 16-bit register as 0FFFFH（all 1）, (CRC register).

b) The first 8-bit byte carries out an “XOR” operation with the low byte of the CRC register;

store the result in the CRC register

c) Shift 1 bit of the CRC register to the right, replace the HSB with “0”, shift out and measure the LSB.

d) If the LSB is 0: repeat the step 3(next shift).

e) If the LSB is 1: carry out the “ XOR ” operation of the CRC register and a preset fixed value（0A001H）.

f) Repeat step c and d until 8digit shift.

g) Repeat step b to step e to handle the next 8-bit until all bytes are processed.

h) The final CRC register value is the CRC.

## Appendix F2 communication application example

F2.1 installation and debugging steps

（1）connect A and B lines of RS485 to 10# and 11# terminals of NA8 ACB secondary circuit respectively.

（2）adjust communication address parameter of controller to 3，baud rate to 9.6kbps according to manual.

Detailed procedures are shown below：

（a）press 

to enter “system set”. Then press  to select “communication set”.

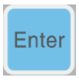
System set
Clock set
Meter set
Test & Lock

System set
Communication set
I/O set

（b）press 

to enter “communication set”;

Communication set
Address = 3
Baud rate = 9.6k

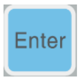
（c）press 

first，then   to adjust communication address to 3. Finally, press 

to save；

（d）press  to select baud rate，then press 

to get ready. Press  

to adjust baud rate to 9.6k and press  to save. (All communication parameter set is finished)

（3）connect RS485 bus convert to debugging computer;

（4）open serial debugging tool（you can download it from internet for free）to set serial number and baud rate. Set serial format as: 8 digit data, no verification and 2 stop bit.

（5）Send test frame 03 03 00 01 00 01 D4 28. If controller sends 03 03 02 00 00 C1 84 back, communication function works well.（Note：A phase current is 0A at the moment）

F2.2 troubleshooting

（1）check if A and B lines of RS485 are connected to 10# and 11# terminals of NA8 ACB secondary circuit reliably.（attention: A->10#，B->11#）；

（2）communication parameter set of controller should be consistent with host computer.

（3）check if serial port setup of debugger is consistent with that of controller.

（4）check if RS485 convert are damaged.(You can try a new one)

（5）Except all above, you can contact our company to have further analysis.