NOJA POWER®

MODBUS Protocol Implementation

Revision History

Rev	Author	Date	Comment
First Release	BOS	26-05-04	First Release
Rev 1	BOS	16-Nov-04	Added I/O points.
Rev 2	OS	05-Jan-10	- Document template changed - Mapping changed - New points added and point numbering changed. The following new points added: Input Coils: 10003, 10006 – 10009, 10149, 10150. - Page 13, Note 1. Frequency multiplication factor documentation error corrected. Old Value: 0.01 when should be 0.001

Contents

1 INTRODUCTION 3
2 MPM FIRMWARE 3
3 IMPLEMENTATION 3
4 TRANSMISSION MODE 3
5 SUPPORTED FUNCTION CODES 4
5.1 Function 01 Read Coil Status 5
5.2 Function 02 Read Input Status 6
5.3 Function 03 Read Holding Registers 10
5.4 Function 04 Read Input Registers 11
5.5 Function 05 Force Single Coil 16
5.6 Function 06 Preset Single Register 16
5.7 Function 15 (0F Hex) Force Multiple Coils 16
5.8 Function 16 (10 Hex) Preset Multiple Registers 16
6 MODBUS EXCEPTION RESPONSES 17

1 INTRODUCTION

The MODBUS Protocol is a messaging structure developed by Modicon in 1979, used to establish master-slave/client-server communication between intelligent devices. It is a de facto standard, truly open and the most widely used network protocol in the industrial manufacturing environment.

The purpose of this document is to describe the specific implementation of the Modicon Modbus protocol on the Recloser Control Cubicle RC-01E(S).

This document, in conjunction with the Modicon Modbus Protocol Reference Guide (PI-MBUS-300) and the MODBUS Application Protocol Specification V1.1, published by Modicon, Inc., provides complete information on how to communicate with the RC-01ES via Modbus

2 MPM FIRMWARE

This document describes the Modbus points list available in firmware versions S02.03.04-6802 and above.

3 IMPLEMENTATION

The RC-01ES supports the Modbus protocol. Modbus can be assigned to either the RS485 or RS232 port and supports transmit/receive data rates of 300, 600, 1200, 2400, 4800, 9600, 14400 and 19200 baud. Half-duplex or full-duplex connections can be used on the RS-485 port. RC-01ES is always slave and supports only RTU (binary) mode.

Valid slave device addresses are in the range of 0 – 247 decimal. The individual slave devices are assigned addresses in the range of 1 – 247. A master addresses a slave by placing the slave address in the address field of the message. When the slave sends its response, it places its own address in this address field of the response to let the master know which slave is responding.

This Modbus implementation supports broadcast message receiving on address 0.

4 TRANSMISSION MODE

THE RC-01ES supports communication on a Modbus network using RTU (Remote Terminal Unit) mode, each 8–bit byte in a message contains two 4–bit hexadecimal characters. The main advantage of this mode is that its greater character density allows better data throughput than ASCII for the same baud rate. Each message must be transmitted in a continuous stream.

Supports format for each byte in RTU mode is:

Coding System:

8–bit binary, hexadecimal 0–9, A–F

Two hexadecimal characters contained in each 8–bit field of the message

Bits per Byte:

1 start bit

8 data bits, least significant bit sent first

1 stop bit

Error Check Field:

Cyclical Redundancy Check (CRC)

Start	1	2	3	4	5	6	7	8	Stop

5 SUPPORTED FUNCTION CODES

The listing below shows the function codes supported by this Modbus implementation. Codes are listed in decimal. 'Yes' indicates that the function is supported. 'No' indicates that it is not supported.

Table 5.1 – Supported function codes

Code	***Name***	***Supported***
01	Read Coil Status	Yes
02	Read Input Status	Yes
03	Read Holding Registers	Yes
04	Read Input Registers	Yes
05	Force Single Coil	Yes
06	Preset Single Register	Yes
07	Read Exception Status	No
08	Diagnostics	No
09	Program 484	No
10	Poll 484	No
11	Fetch Comm. Event Ctr.	No
12	Fetch Comm. Event Log	No
13	Program Controller	No
14	Poll Controller	No
15	Force Multiple Coils	Yes
16	Preset Multiple Registers	Yes
17	Report Slave ID	No
18	Program 884/M84	No
19	Reset Comm. Link	No
20	Read General Reference	No
21	Write General Reference	No
22	Mask Write 4X Register	No
23	Read/Write 4X Registers	No
24	Read FIFO Queue	No

5.1 Function 01 Read Coil Status

5.1.1 Description

Reads the ON/OFF status of discrete outputs (0X references, coils) in the slave.
Broadcast is not supported.

The query message specifies the starting coil and quantity of coils to be read.
Coils are addressed starting at zero: coils 1–16 are addressed as 0–15.

Table 5.2 - 0X references, output coils

Address	Settings	Description
00001	Trip	Trip (Note: A Trip operation is also possible when the control mode is set to local)
00002	Close	Close
00003	On (Grp1)	Switch Group1 On
00004	On (Grp2)	Switch Group2 On
00005	On (Grp3)	Switch Group3 On
00006	On (Grp4)	Switch Group4 On
00007	On (Prot)	Switch protection On
00008	Off (Prot)	Switch protection Off
00009	On (EF)	Switch earth fault overcurrent element On
00010	Off (EF)	Switch earth fault overcurrent element Off
00011	On (SEF)	Switch sensitive earth fault element On
00012	Off (SEF)	Switch sensitive earth fault element Off
00013	On (AR)	Switch autoreclosing element On
00014	Off (AR)	Switch autoreclosing element Off
00015	On (LL)	Switch live line element On
00016	Off (LL)	Switch live line element Off
00017	On (CLP)	Switch cold load pickup element On
00018	Off (CLP)	Switch cold load pickup element Off
00019	On (UV)	Switch undervoltage element On
00020	Off (UV)	Switch undervoltage element Off
00021	On (UF)	Switch underfrequency load shed element On
00022	Off (UF)	Switch underfrequency load shed element Off
00023	On (ABR)	Switch ABR element On
00024	Off (ABR)	Switch ABR element Off
00025-00030	On/Off (IO1, Outputs 1-6)	Switch IO1 Module (Outputs 1-6) On/Off
00031-00036	On/Off (IO2, Outputs 1-6)	Switch IO2 Module (Outputs 1-6) On/Off
00037	Reserved	
...	...	...
00048	Reserved	

NOTE: Reserved – This coil is always 0.

The coil status in the response message is packed as one coil per bit of the data field. Status is indicated as: 1 = ON; 0 = OFF.

5.5 Function 05 Force Single Coil

5.5.1 Description

Forces a single coil (0X reference) to either ON or OFF. When broadcast, the function forces the same coil reference in all attached slaves.

If an error occurs in the processing of a broadcast request, the exception response is not formed.

The setting of coils does not occur with the appearance of exception.

The list of coils is given in Table 5.2.

NOTE 1: The setting of coil in 0 will not result in any changes in the RC-01E(S).
NOTE 2: The setting of "Reserved" coil in 0 or 1 will not result in any changes in the RC-01E(S).

5.6 Function 06 Preset Single Register

5.6.1 Description

Presets a value into a single holding register (4X reference). When broadcast, the function presets the same register reference in all attached slaves.

If there is an error in the processing of a broadcast request, the exception response is not formed.

The list of coils is given in Table 5.4.

5.7 Function 15 (0F Hex) Force Multiple Coils

5.7.1 Description

Forces each coil (0X reference) in a sequence of coils to either ON or OFF. When broadcast, the function forces the same coil references in all attached slaves.

If there is an error in the processing of a broadcast request, the exception response is not formed.

The setting of coils does not occur with the appearance of exception.

The list of coils is given in Table 5.2.

NOTE 1: The setting of coil in 0 doesn’t affect the protection algorithms in the RC-01E(S).
NOTE 2: The setting of "Reserved" coil in 0 or 1 will not result in any changes in the RC-01E(S).

5.8 Function 16 (10 Hex) Preset Multiple Registers

5.8.1 Description

Presets values into a sequence of holding registers (4X references). When broadcast, the function presets the same register references in all attached slaves.

If there is an error in the processing of a broadcast request, the exception response is not formed.

The list of coils is given in Table 5.4.

6 MODBUS EXCEPTION RESPONSES

The exception response message has two fields that differentiate it from a normal response:

Function Code Field: In a normal response, the server echoes the function code of the original request in the function code field of the response. All function codes have a most–significant bit (MSB) of 0 (their values are all below 80 hexadecimal). In an exception response, the server sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response.

With the function code's MSB set, the client's application program can recognize the exception response and can examine the data field for the exception code.

Data Field: In a normal response, the server may return data or statistics in the data field (any information that was requested in the request). In an exception response, the server returns an exception code in the data field, defining the server condition that caused the exception.

This Modbus implementation returns three types of errors; they are given in table 5.1. The complete list of possible errors are documented in the PI-MBUS-300 document.

Table 6.1 - MODBUS Exception Codes

MODBUS Exception Codes		
Code	***Name***	***Meaning***
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the slave. If a Poll Program Complete command was issued, this code indicates that no program function preceded it. This error can arise with the demand of the non supported function
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the slave. This error can arise in such a case when a nonexistent data object address is used in the request.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the slave. This error can arise during an attempt to set values at the moment when the RC-01E(S) is set in the Remote Off (Local) mode.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the slave was attempting to perform the requested action.