Modicon
Modbus Protocol
Reference Guide

PI–MBUS–300 Rev. J

June 1996

MODICON, Inc., Industrial Automation Systems
One High Street
North Andover, Massachusetts 01845

Contents

Introducing Modbus Protocol (Continued)

Transactions on Modbus Networks

Standard Modbus ports on Modicon control,s use an RBUSò2Con mpatible serialks

At the message level, the Modbus protocol still applies the master–slave principle even though the network communication method is peer–to–peer. If a controller

The Two Serial Transmission Modes

RTU Mode

When controllers are setup to communicate 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 dan.ooejdeMh is n ASCII for is arame baud rgre.ter

Modbus Message Framing

In either of the two serial transmission modes (ASCII or RTU), a Modbus message is placed by the transmitting device into a frame that has a known beginning and

Exception: With the 584 and 984A/B/X controllers, an ASCII message can normally terminate after the LRC field without the CRLF characters being sent. An interval of at least one second must then occur. If this happens, the controller will assume that the message terminated normally.

RTU Framing

In RTU mode, messages start with a silent interval of at least 3.5 character times. This is most easily implemented as a multiple of character times at the baud rate that is being used on the network (shown as T1–T2–T3–T4 in the figure below). The first field then transmitted is the device address.

The allowable characters transmitted for all fields are hexadecimal 0–9, A–F.

If the slave device takes the requested action without error, it returns the same code in its response. If an exception occurs, it returns:

1000 0011 (Hexadecimal 83)funested 1 TwÏforse. If an excin its r,mÏÈ[eturne

How Characters are Transmitted Serially

When messages are transmitted on standard Modbus serial networks, each character or byte is sent in this order (left to right):

Least Significant Bit (LSB) . . . Most Significant Bit (MSB)

With ASCII character framing, the bit sequence is:

Error Checking Methods

Error Checking Methods (Continued)

CRC Checking

In RTU mode, messages include an error–checking field that is based on a Cyclical Redundancy Check (CRC) method. The CRC field checks the contents of the entire message. It is applied regardless of any parity check method used for the individual characters of the message.

The CRC field is two bytes, containing a 16–bit binary value. The CRC value is calculated by the transmitting device, which appends the CRC to the message. The receiving device recalculates a CRC during receipt of the message, and compares the calculated value to the actual value it received in the CRC field. If the two values are not equal, an error results.

The CRC is started by first preloading a 16–bit register to all 1's. Then a process begins of applying successive 8–bit bytes of the message to the current contents of the register. Only the eight bits of data in each character are used for generating the CRC. Start and stop bits, and the parity bit, do not apply to the CRC.

During generation of the CRC, each 8–bit character is exclusive ORed with the register contents. Then the result is shifted in the direction of the lea inorniflicnts LSB),d witha zeroy fllted i to the oa inorniflicnt 1bit(MSB) posictioe. TheLSBe is
If theLSBewaes a1,f the registed is Then exclusive ORes
If theLSBewaes a0,o nn exclusive O taktesplacge.

(eighh) shif,f thenexte 8–bit bytr is exclusive ORed with the register-38(')20(s)0(e currena)]TJÏÈ0 -1.3228 TDÏÈ-0.003

fllowted by theheig-ordter bytd.

Inladdterlogic,f theCKSM funectionecalculates a CRCfromf the message contentse comucters, sdentaltedexamples of CRC generation is

Chapter 2
Data and Control Functions

Field Contentunon Modbus Plus

Function Codes Supported by Controllers

Code	Name	384	484	584	884	M84	984
22	Mask Write 4X Register	N	N	N	N	N	(1)
23	Read/Write 4X Registers	N	N	N	N	N	(1)
24	Read FIFO Queue	N	N	N	N	N	(1)

Notes:

(1) Function is supported in 984–785 only.

01 Read Coil Status

Description

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

02 Read Input Status

Description

Reads the ON/OFF status of discrete inputs (1X references) in the slave.
Broadcast is not supported.

Appendix B lists the maximum parameters supported by vari5tc Contrler models.00

03 Read Holding Registers

Description

Reads the binary contents of holding registers (4X references) in the slave. Broadcast is not supported.

Appendix B lists the maximum parameters supported by various controller models.

Query

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

Here is an example of a request to read registers 40108–40110 from slave device 17:

QUERY

Field Name	Example (Hex)
Slave Address	11
Function	03
Starting Address Hi	00
Starting Address Lo	6B
No. of Points Hi	00
No. of Points Lo	03
Error Check (LRC or CRC)	—

Figure 14 Read Holding Registers – Query

Response

05 Force Single Coil

Description

Response

The normal response is an echo of the query, returned after the register contents have been preset.

Here is an example of a response to the query on the opposite page:

RESPONSE

Field Name	Example (Hex)
Slave Address	11
Function	06
Register Address Hi	00
Register Address Lo	01
Preset Data Hi	00
Preset Data Lo	03
Error Check (LRC or CRC)	—

Figure 21 Preset Single Register – Response

07 Read Exception Status

Description

Reads the contents of eight Exception Status coils within the slave controller. Certain coils have predefined assignments in the various controllers. Other coils can be programmed by the user to hold information about the contoller's status, for example, 'machine ON/OFF', 'heads retracted', 'safeties satisfied', 'error conditions exist', or other user–defined flags. Broadcast is not supported.

The function provides a simple method for accessing this information, because the Exception Coil references are known (no coil reference is needed in the function). The predefined Exception Coil assignments are:

Controller Model	Coil	Assignment
M84, 184/384, 584, 984	1 – 8	User defined
484	257	Battery Status
	258 – 264	User defined
884	761	Battery Status
	762	Memory Protect Status
	763	RIO Health Status
	764–768	User defined

Query

Here is an examplethe Bi:tecasr tr Rean the Exceptio1(stated ie slavdevienc17re:)Tj8È9 0 8 277.9.00304.44129 TmÏE

Response

The normal response contains the status of the eight Exception Status coils. The coils are packed into one dÈ0 gyte, witho onebit pere cois. Tthe status of th

Response

The normal response contains a two–byte status word, and a two–byte event

Response

The normal response contains a two–byte status word field, a two–byte event count field, a two–byte message count field, and a field containing 0-64 bytes of events. A byte count field defines the total length of the data in these four fields.

Here is an example of a response to the query on the opposite page:

Field Name	Example (Hex)
Slave Address	11
Function	0C

6 1
7 0

Slave Entered Listen Only Mode

This type of event byte is stored by the slave when it enters the Listen Only Mode.
70

15 (0F Hex) Force Multiple Coils

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.

> **Note** The function will override the controller's memory protect state and a coil's disable state. The forced state will remain valid until the controller's logic next solves each coil. Coils will remain forced if they are not programmed in the controller's logic.

Appendix B lists the maximum parameters supported by various controller models.

Query

The query message specifies the coil references to be forced. Coils are addressed starting at zero: coil 1 is addressed as 0.

The reguested ON/OFF states are specified by contents of the query data field. A logical '1' in a bit position of the field requests the corresponding coil to be ON. A logical '0' requests it to be OFF.

The following page shows an example of a request to force a series of ten coils starting at coil 20 (addressed as 19, or 13 hex) in slave device 17.

The query data contents are two bytes: CD 01 hex (1100 1101 0000 0001 binary). The binary bits correspond to the coils in the following way:

Bit:	1	1	0	0	1	1	0	1		0	0	0	0	0	0	0	1
Coil:	27	26	25	24	23	22	21	20		–	–	–	–	–	–	29	28

The first byte transmitted (CD hex) addresses coils 27-20, with the least significant bit addressing the lowest coil (20) in this set.

The next byte transmitted (01 hex) addresses coils 29-28, with the least significant bit addressing the lowest coil (28) in this set. Unused bits in the last data byte should be zero–filled.

16 (10 Hex) Preset Multiple Registers

Description

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

> **Note** The function will override the controller's memory protect state. The preset values will remain valid in the registers until the controller's logic next solves the register contents. The register values will remain if they are not programmed in the controller's logic.

Appendix B lists the maximum parameters supported by various controller models.

Query

The query message specifies the register references to be preset. Registers are

17 (11 Hex) Report Slave ID (Continued)

184/384

8, 9 Machine stop code (configuration table word 105, 69 hex). The word is organized as follows:

Byte 8:
Bit 15 (MSB of byte 8) = Peripheral port stop (controlled stop)
Bit 14 = Unassigned
Bit 13 = Dim awareness

17 (11 Hex) Report Slave ID (Continued)

Micro 84

The Micro 84 controller returie5 170re9]rig of 8, as frolows:0

884

The 884 controller returns a byte count of 8, as follows:

Byte	Contents
1	Slave ID (8 for 884)
2	RUN indicator status (0 = OFF
	izet ofuslerlogic pltus stae RAM,N i kilo bytus Â word0 =2a byts4)
52	esleved

20 (14Hex) Read General Reference

Description

Returns the contents of registers in Extended Memory file (6XXXXX) references.

20 (14 Hex) Read General Reference (Continued)

An example of a request to read two groups of references from slave device 17 is shown below.

Group 1 consists of two registers from file 4, starting at register 1 (address 0001).
Group 2 consists of two registers from file 3, starting at register 9 (address 0009).

QUERY Field Name	Example (Hex)
Slave Address	11
Function	14
Byte Count	0E
Sub–Req 1, Reference Type	06
Sub–Req 1, File Number Hi	00
Sub–Req 1, File Number Lo	04
Sub–Req 1, Starting Addr Hi	00
Sub–Req 1, Starting Addr Lo	01
Sub–Req 1, Register Count Hi	00
Sub–Req 1, Register Count Lo	02
Sub–Req 2, Reference Type	06
Sub–Req 2, File Number Hi	00
Sub–Req 2, File Number Lo	03
Sub–Req 2, Starting Addr Hi	00
Sub–Req 2, Starting Addr Lo	09
Sub–Req 2, Register Count Hi	00
Sub–Req 2, Register Count Lo	02
Error Check (LRC or CRC)	—

Figure 34 Read General Reference – Query

21 (15Hex) Write General Reference

Description

Writes the contents of registers in Extended Memory file (6XXXXX) references. Broadcast is not supported.

The function can write multiple groups of references. The groups can be separate (non–contiguous), but the references within each group must be sequential.

Query

The query contains the standard Modbus slave address, function code, byte count, and error check fields. The rest of the query specifies the group or groups of references to be written, and the data to be written into them. Each group is defined in a separate ‘sub-request’ field which contains 7 bytes plus the data:

- The reference type: 1 byte (must be specified as 6)
- The Extended Memory file number: 2 bytes (1 to 10, hex 0001 to 000A)
- The starting register address within the file: 2 bytes
- The quantity of registers to be written: 2 bytes
- The data to be written: 2 bytes per register.

The quantity of registers to be written, combined with all other fields in the query, must not exceed the allowable length of Modbus messages: 256 bytes.

The available quantity of Extended Memory files depends upon the installed size of Extended Memory in the slave controller. Each file except the last one contains 10,000 registers, addressed as 0000-270F hexadecimal (0000-9999 decimal).

> **Note** The addressing of Extended Register (6XXXXX) references differs from that of Holding Register (4XXXX) references.
>
> The lowest Extended Register is addressed as register ‘zero’ (600000).

For controllers other than the 984–785 with Extended Registers, the last (highest) register in the last file is:

96K	**10**	**83063**

QUERY

Field Name	Example (Hex)
Slave Address	11
Function	16
Reference Address Hi	00
Reference Address Lo	04
And_Mask Hi	00
And_Mask Lo	F2
Or_Mask Hi	00
Or–Mask Lo	25
Error Check (LRC or CRC)	—

Figure 38 Mask Write 4X Register – Query

Response

The normal response is an echo of the query. The response is returned after the register has been written.

RESPONSE

Field Name	Example (Hex)
Slave Address	11
Function	16
Reference Address Hi	00
Reference Address Lo	04
And_Mask Hi	00
And_Mask Lo	F2
Or_Mask Hi	00
Or–Mask Lo	25
Error Check (LRC or CRC)	—

Figure 39 Mask Write 4X Register – Response

Response

Chapter 3
Diagnostic Subfunctions

- Modbus Function 08 – Diagnostics
- Diagnostic Subfunctions

Query

Here is an example of a request to slave device 17 to Return Query Data. This

08 Diagnostics (Continued)

02 Return Diagnostic Register

03 Change ASCII Input Delimiter

The character ‘CHAR’ passed in the query data field becomes the end of message delimiter for future messages (replacing the default LF character). This function is

Modbus Plus Network Statistics

Word	Bits	Meaning
00		Node type ID:
	0	Unknown node type
	1	Programmable controller node
	2	Modbus bridge node
	3	Host computer node
	4	Bridge Plus node
	5	Peer I/O node
01	0 ... 11	Software version number in hex (to read, strip bits 12–15 from word)
	12 ... 14	Reserved
	15	Defines Word 15 error counters (see Word 15)

Most significant bit defines use of465 TDÏÈ-0.01 Tw.o W

Word	Byte	Meaning
18	LO	Bad internal packet length error counter
	HI	Bad MAC function code error counter
19	LO	Communication retry counter
	HI	Communication failed error counter
20	LO	Good receive packet success counter
	HI	No response received error counter
21	LO	Exception response received error counter
	HI	Unexpected path error counter
22	LO	Unexpected response error counter
	HI	Forgotten transaction error counter
23	LO	Active station table bit map, nodes 1 ... 8
	HI	Active station table bit map, nodes 9 ...16
24	LO	Active station table bit map, nodes 17 ... 24
	HI	Active station table bit map, nodes 25 ... 32
25	LO	Active station table bit map, nodes 33 ... 40
	HI	Active station table bit map, nodes 41 ... 48
26	LO	Active station table bit map, nodes 49 ... 56
	HI	Active station table bit map, nodes 57 ... 64
27	LO	Token station table bit map, nodes 1 ... 8
	HI	Token station table bit map, nodes 9 ... 16
28	LO	Token station table bit map, nodes 17 ... 24
	HI	Token station table bit map, nodes 25 ... 32
29	LO	Token station table bit map, nodes 33 ... 40
	HI	Token station table bit map, nodes 41 ... 48
30	LO	Token station table bit map, nodes 49 ... 56
	HI	Token station table bit map, nodes 57 ... 64
31	LO	Global data present table bit map, nodes 1 ... 8
	HI	Global data present table bit map, nodes 9 ... 16
32	LO	Global data present table bit map, nodes 17 ... 24
	HI	Global data present table bit map, nodes 25 ... 32
33	LO	Global data present table bit map, nodes 33 ... 40
	HI	Global data present table bit map, nodes 41 ... 48
34	LO	Global data present table map, nodes 49 ... 56
	HI	Global data present table bit map, nodes 57 ... 64

Modbus Plus Network Statistics (Continued)

Word	Bits	Meaning
35	LO	Receive buffer in use bit map, buffer 1–8
	HI	Receive buffer in use bit map, buffer 9 ... 16
36	LO	Receive buffer in use bit map, buffer 17 ... 24
	HI	Receive buffer in use bit map, buffer 25 ... 32
37	LO	Receive buffer in use bit map, buffer 33 ... 40
	HI	Station management command processed initiation counter
37	LO	
	HI37	LO
		HI37 LO
		HI37 LO
		HI
		37 LOReinput pathT41gement command procation counter
		HIReinput pathT42gement command procation counter37 LOReinput pathT4
		HIReinput pathT44gement command procation counter37 LORe
		HIReinput pathT46gement command procation counter35
		HIReinput pathT48gement command procation c
		36 LO
		HI37 LO
		HI

Appendix A
Exception Responses

- Exception Responses
- Exception Codes

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

Appendix B
Application Notes

This Appendix contains information and suggestions for using Modbus in your application.

- Maximum Query/Response Parameters for Modicon Controllers

  Estimating Serial Transaction Timing

  Application Notes for the 584 and 984A/B/X Controllers

Maximum Q/R Parameters (Continued)

584

Function	Description	Query	Response
1	Read Coil Status	2000 coils	2000 coils
2	Read Input Status	2000 inputs	2000 inputs
3	Read Holding Registers	125 registers	125 registers
4	Read Input Registers	125 registers	125 registers

884

Function	Description	Query	Response
1	Read Coil Status	2000 coils	2000 coils

Maximum Q/R Parameters (Continued)

(C) Continued:

For 484 controllers the time is approximately 1.5 ms. The Modbus port is available on a contention basis with any J470/J474/J475 that is present.

For 584 and 984 controllers the time is approximately 1.5 ms for each Modbus port. The ports are serviced sequentially, starting with port 1.

For 184/384 controllers the time varies according to the amount of data being handled. It ranges from a minimum of 0.5 ms to a maximum of about 6.0 ms (for 100 registers), or 7.0 ms (for 800 coils). If a programming panel

Notes for the 584 and 984A/B/X

These application notes apply only to Modicon 584 and 984A/B/X controllers.

Baud Rates: When using both Modbus ports 1 and 2, the maximum allowable *combined* baud rate is 19,200 baud.

Port Lockups: When using ASCII, avoid sending 'zero data length' messages, or messages with no device address. For example, this is an illegal message:

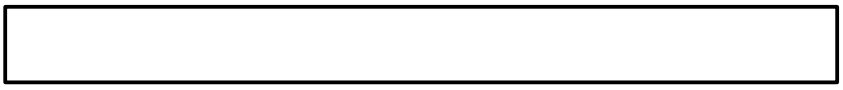
: CR LF (colon, CR, LF)

Random port lockups can occur this kind of message is used.

Terminating ASCII Messages: ASCII messages should normally terminate with a CRLF pair. With the 584 and 984A/B/X controllers, an ASCII message can terminate after the LRC field (without the CRLF characters being sent), if an interval of at least one second is allowed to occur after the LRC field. If this

Appendix C
LRC/CRC Generation

- LRC Generation
- CRC Generation



CRC Generation

CRC Generation (Continued)

Example (Continued)

The function takes two arguments:

```
unsigned char *puchMsg ;
```

High-Order Byte Table