Modbus Protocol Reference Guide

Contents

1. Introduction

The Modbus protocol is provided by Modicon Inc. (AEG Schneider Automation International S.A.S.), originally developed for Modicon programmable controllers. Detailed information is described in Modicon Modbus Protocol Reference Guide (PI-MBUS-300 Rev. J).

This protocol defines a message structure, regardless of the physical layer such like the type of networks over which they communicate.

2. Modbus Protocol

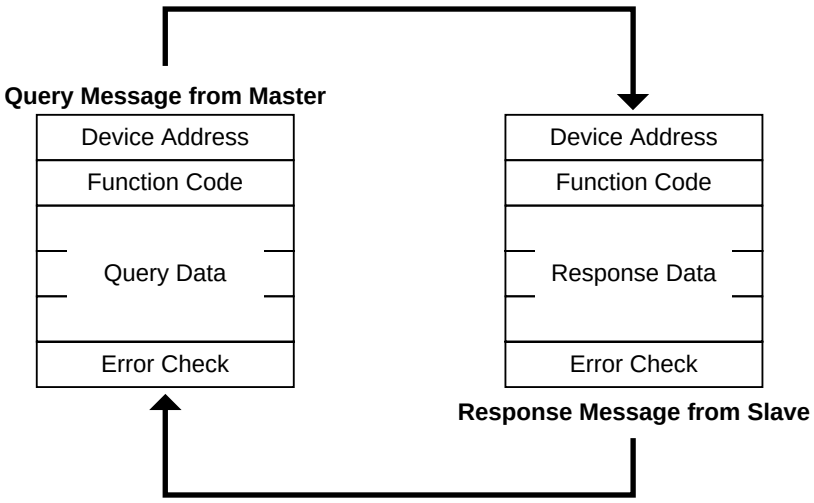
2.1 General Description

The Modbus devices communicate using a master-slave technique, in which only one device (the master) can initiate transactions (called 'queries'). The other devices (the slaves) respond by supplying the requested data to the master, or by taking the action requested in the query.

The master can address individual slaves, or can initiate a broadcast messages to all slaves. Slaves return a message (called a 'response') to queries that are addressed to them individually. Responses are not returned to broadcast queries from the master.

The Modbus protocol establishes the format for the master's query by placing into it the device (or broadcast) address, a function code defining the requested action, any data to be sent, and an error-checking field. The slave's response message is also constructed using Modbus protocol. It contains fields confirming the action taken, any data to be returned, and an error-checking field.

The figure below illustrates a query-response cycle.



Devices can be setup to communicate on standard Modbus networks either of two transmission modes: ASCII (American Standard Code for Information Interchange) or RTU (Remote Terminal Unit). The mode must be the same for all devices on a Modbus network.

In ASCII mode, each 8-bit byte in a message is sent as two ASCII characters. In RTU mode, each 8-bit byte in a message contains two 4-bit hexadecimal characters. The RTU mode, with its greater character density, allows better data throughput than ASCII for the same baud rate.

The checking algorithm used in the Error Check Field depends upon which transmission method is employed; LRC (Longitudinal Redundancy Check) in ASCII mode; CRC (Cyclical Redundancy Check) in RTU mode.

2.2 Modbus Message Framing

ASCII Framing

In ASCII mode, messages start with a 'colon' (:) character (ASCII 0x3A), and end with a 'carriage return – line feed' (CRLF) pair (ASCII 0x0D and 0x0A). The allowable characters transmitted for all other fields are hexadecimal 0 – 9, A – F.

When the messages are to be sent over Ethernet, this message frame is handled as a data frame in TCP/IP protocol. Dividing a message frame is not allowed.

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	*n* CHARS	2 CHARS	2 CHARS CRLF

Figure 1. ASCII Message Frame

RTU Framing

In RTU mode, messages start with a silent interval of at least 3.5 character times, and end with a similar 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).

All other fields are composed of 8-bit data.

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1–T2–T3–T4*	8 BITS	8 BITS	*n* x 8 BITS	16 BITS	T1–T2–T3–T4*

*For T1–T2–T3–T4, 3.5 character times at no communication.

Figure 2. RTU Message Frame

Address Field

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. Address 0 is used for the broadcast query.

Function Field

Valid function field codes are in the range of 1 – 255 decimal.

When a message is sent from a master to a slave device the function code field tells the slave what kind of action to perform. When the slave responds to the master, it uses the function code field to indicate either a normal (error-free) response or that some kind of error occurred (called an exception response). For a normal response, the slave simply echoes the original function code. For an exception response, the slave returns a code that is equivalent to the original function code with its most-significant bit set to a logic 1. This tells the master what kind of error occurred, or the reason for the exception.

Whether a particular function code is applicable or not depends upon the slave device. Check specifications for each slave device.

Data Field

The data field of messages sent from a master to slave devices contains information which the slave must use to take the action defined by the function code. The data field may be of various length, or can be nonexistent (of zero length). Refer to specifications for each slave device for the constructions and meaning of the data field.

Error Checking Field

ASCII

When ASCII mode is used for character framing the error checking field contains two ASCII characters. The error check characters are the result of a Longitudinal Redundancy Check (LRC) calculation that is performed on the message content, exclusive of the beginning 'colon' and terminating CRLF characters.

RTU

When RTU mode is used for character framing, the error checking field contains a 16-bit value implemented as two 8-bit bytes. The error check value is the result of a Cyclical Redundancy Check calculation performed on the message contents.

LRC Checking

In ASCII mode, messages include an error-checking field that is based on a Longitudinal Redundancy Check (LRC) method. The LRC field checks the contents of the message, exclusive of the beginning 'colon' and ending CRLF pair. It is applied regardless of any parity check method used for the individual characters of the message.

The LRC field is one byte, containing an 8-bit binary value. The LRC value is calculated by the transmitting device, which appends the LRC to the message. The receiving device calculates an LRC during receipt of the message, and compares the calculated value to the actual value it received in the LRC field. If the two values are not equal, an error results.

The LRC is calculated by adding together successive 8-bit bytes of the message, discarding any carries, and then two's complementing the result. It is performed on the ASCII message field contents excluding the 'colon' character that begins the message, and excluding the CRLF pair at the end of the message.

E.g. 1

The query that reads the resistor 30001 in Slave device 1 is following. (Refer to 3.4 about query message)

":", "0", "1", "0", "4", "0", "0", "0", "0", "0", "0", "0", "1", "F", "A", CR/LF
For the above query message, LRC is "F", "A."

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 to generating the CRC. Start and stop bits, and the parity bit if one is used, 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 least significant bit (LSB), with a zero filled into the most significant bit (MSB) position. The LSB is extracted and examined. If the LSB was a 1, the register is then exclusive ORed with a preset, fixed value. If the LSB was a 0, no exclusive OR takes place.

This process is repeated until eight shifts have been performed. After the last (eighth) shift, the next 8-bit byte is exclusive ORed with the register's current value, and the process repeats for eight more shifts as described above. The final contents of the register, after all the bytes of the message have been applied, is the CRC value.

E.g. 2

The query that reads the resistor 30001 in Slave device 1 is following. (Refer to 3.4 about query message)

0x01, 0x04, 0x00, 0x00, 0x00, 0x01, 0x31, 0xCA
For the above query message, the calculated value of CRC is 0xCA31. The lower-order byte in CRC is appended first, consequently the message order is 0x31, 0xCA.

When the CRC is appended to the message, the low-order byte of the field is appended first, followed by the high-order byte.

2.3 Modbus Function Formats

Data Address

Data addresses are used in Modbus query messages when reading or modifying data. Four types of data are used: Coil, Input Status, Input Register and Holding Register.

Coil

Coils are used to force the ON/OFF state of discrete outputs (DO) to the field, or to modify the mode or status of slave devices. Coil data is either ON or OFF, which can be both read and modified. Valid addresses are in the range of 1 – 9999.

Input Status

Input Status is used for the ON/OFF state of discrete inputs (DI) from the field, or the status of slave devices. The input status is either ON or OFF, which can be read only. Valid addresses are in the range of 10001 – 19999.

Input Register

Input registers are used for the value of analog inputs (AI) from the field, or the information of slave devices. The input register is of 16-bit long, which can be read only. Valid addresses are in the range of 30001 – 39999. Floating or double-floating data can be handled when consecutive addresses are assigned.

Holding Register

Holding registers are used for the value of analog outputs (AO) to the field, or to set information of slave devices. The holding register is of 16-bit long, which can be both read and modified. Valid addresses are in the range of 40001 – 49999. Floating or double-floating data can be handled when consecutive addresses are assigned.

2.4 Field Contents in Modbus Messages

All data addresses in Modbus messages are referenced to 0. The first occurrence of a data item is addressed as item number zero. For example, the input register 30156 decimal is addressed as register 155 decimal in the message field. The function code field specifies data type.

Figure 3 shows an example of a Modbus query message. The master query is a Read Holding Registers request (function code 03) to slave device address 06. The message requests data from three holding registers, 40108 through 40110. Note that the messages specifies the starting register address as 107 (0x6B hex).

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x06	0 6	0x06
Function	0x03	0 3	0x03
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x6B	6 B	0x6B
No. of Registers Hi	0x00	0 0	0x00
No. of Registers Lo	0x03	0 3	0x03
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 3. Example of Master Query

Figure 4 is an example of a normal response from the slave to the master query shown in Figure 3. The slave response echoes the slave address and function code. The 'Byte Count' field specifies how many 8-bit data items are being returned. Note that the value does not represent the actual character count transmitted in either ASCII or RTU mode. In this example, the message contains three sets of 16-bit data, therefore the 'Byte Count' is '6' regardless of the character framing method.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x06	0 6	0x06
Function	0x03	0 3	0x03
Byte Count	0x06	0 6	0x06
Data 1 Hi	0x03	0 3	0x03
Data 1 Lo	0xE8	E 8	0xE8
Data 2 Hi	0x01	0 1	0x01
Data 2 Lo	0xF4	F 4	0xF4
Data 3 Hi	0x00	0 0	0x00
Data 3 Lo	0x0A	0 A	0x0A
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	23	11

Figure 4. Example of Slave Response

3. Modbus Function Codes

3.1 Read Coil Status (01)

Description

Read the ON/OFF status of discrete outputs (DO) in the slave. Broadcast is not supported. Refer to specifications of the slave device for data addresses and their contents.

Query

The query message specifies the starting coil and quantity of coils to be read. Here is an example of a request to read coils 20 – 56, 37 coils in total, from slave device 3: (Note that the Starting Address is of 19 or 0x13, less than the coil 20 by 1.)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x03	0 3	0x03
Function	0x01	0 1	0x01
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x13	1 3	0x13
No. of Registers Hi	0x00	0 0	0x00
No. of Registers Lo	0x25	2 5	0x25
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 5. Read Coil Status – Query

Response

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. The LSB of the first data byte contains the coil addressed in the query.

For example, when the status of coils 20 – 27 is shown ON – ON – OFF – OFF – ON – OFF – ON – OFF, represented as the byte value binary 0101 0011 (0x53). One byte contains the status for eight coils. If the coil quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros.

Figure 6 shows an example of a response to the query shown in Figure 5.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x03	0 3	0x03
Function	0x01	0 1	0x01
Byte Count	0x05	0 5	0x05
Data 1	0x53	5 3	0x53
Data 2	0x6B	6 B	0x6B
Data 3	0x01	0 1	0x01
Data 4	0xF4	F 4	0xF4
Data 5	0x1B	1 B	0x1B
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	21	10

Figure 6. Read Coil Status – Response

3.2 Read Input Status (02)

Description

Read the ON/OFF status of discrete inputs (DI) in the slave. Broadcast is not supported. Refer to specifications of the slave device for data addresses and their contents.

Query

The query message specifies the starting input and quantity of inputs to be read. Here is an example of a request to read inputs 10101 – 10120, 20 inputs in total, from slave device 3: (Note that the Starting Address is of 100 or 0x64, less than the input 10101 by 10001.)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x03	0 3	0x03
Function	0x02	0 2	0x02
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x64	6 4	0x64
No. of Registers Hi	0x00	0 0	0x00
No. of Registers Lo	0x14	1 4	0x14
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 7. Read Input Status – Query

Response

The construction of the response message is the same as that for Real Coil Status (01) operation.

Figure 8 shows an example of a response to the query shown in Figure 7.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x03	0 3	0x03
Function	0x02	0 2	0x02
Byte Count	0x03	0 3	0x03
Data 1	0x53	5 3	0x53
Data 2	0x6B	6 B	0x6B
Data 3	0x01	0 1	0x01
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 8. Read Input Status – Response

3.3 Read Holding Register (03)

Description

Read the binary contents of holding registers in the slave. Broadcast is not supported. Refer to specifications of the slave device for data addresses and their contents.

Query

The query message specifies the starting register and quantity of registers to be read. Here is an example of a request to read registers 40201 – 40203, 3 registers in total, from slave device 7: (Note that the Starting Address is of 200 or 0xC8, less than the register 40201 by 40001.)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x07	0 7	0x07
Function	0x03	0 3	0x03
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0xC8	C 8	0xC8
No. of Registers Hi	0x00	0 0	0x00
No. of Registers Lo	0x03	0 3	0x03
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 9. Read Holding Register – Query

Response

The register data in the response message are packed as 16 bits per register. Figure 10 shows an example of a response to the query shown in Figure 9.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x07	0 7	0x07
Function	0x03	0 3	0x03
Byte Count	0x06	0 6	0x06
Data 1 Hi	0x03	0 3	0x03
Data 1 Lo	0xE8	E 8	0xE8
Data 2 Hi	0x01	0 1	0x01
Data 2 Lo	0xF4	F 4	0xF4
Data 3 Hi	0x00	0 0	0x00
Data 3 Lo	0x0A	0 A	0x0A
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	23	11

Figure 10. Read Holding Register – Response

3.4 Read Input Register (04)

Description

Read the binary contents of input registers in the slave. Broadcast is not supported. Refer to specifications of the slave device for data addresses and their contents.

Query

The query message specifies the starting register and quantity of registers to be read. Here is an example of a request to read registers 30301 – 30303, 3 registers in total, from slave device 7: (Note that the Starting Address is of 300 or 0x12C, less than the register 30301 by 30001.)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x07	0 7	0x07
Function	0x04	0 4	0x04
Starting Address Hi	0x01	0 1	0x01
Starting Address Lo	0x2C	2 C	0x2C
No. of Registers Hi	0x00	0 0	0x00
No. of Registers Lo	0x03	0 3	0x03
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 11. Read Input Register – Query

Response

The register data in the response message are packed as 16 bits per register. Figure 12 shows an example of a response to the query shown in Figure 11.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x07	0 7	0x07
Function	0x04	0 4	0x04
Byte Count	0x06	0 6	0x06
Data 1 Hi	0x03	0 3	0x03
Data 1 Lo	0xE8	E 8	0xE8
Data 2 Hi	0x01	0 1	0x01
Data 2 Lo	0xF4	F 4	0xF4
Data 3 Hi	0x00	0 0	0x00
Data 3 Lo	0x0A	0 A	0x0A
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	23	11

Figure 12. Read Input Register – Response

3.5 Force Single Coil (05)

Description

Forces a single coil to either ON or OFF of the discrete output (DO) status in the slave. When broadcast, the function forces the same coil reference in all attached slaves. Refer to specifications of the slave device for data addresses and their contents.

Query

The query message specifies the coil reference (the starting coil and the state) to be forced. The requested ON/OFF state is specified by a constant in the query data field. A value of 0xFF, 0x00 requests the coil to be ON. A value of 0x00, 0x00 requests it to be OFF. All other values are illegal and will not affect the coil.

Here is an example of a request to force coil 150 ON in slave device 3: (Note that the Starting Address is of 149 or 0x95, less than the force coil 150 by 1.)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x03	0 3	0x03
Function	0x05	0 5	0x05
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x95	9 5	0x95
No. of Registers Hi	0xFF	F F	0xFF
No. of Registers Lo	0x00	0 0	0x00
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 13. Force Single Coil – Query

Response

The normal response is an echo of the query. Figure 14 shows an example of a response to the query shown in Figure 13.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x03	0 3	0x03
Function	0x05	0 5	0x05
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x95	9 5	0x95
No. of Registers Hi	0xFF	F F	0xFF
No. of Registers Lo	0x00	0 0	0x00
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 14. Force Single Coil – Response

3.6 Preset Single Register (06)

Description

Presets a value into a single holding register. When broadcast, the function presets the same register reference in all attached slaves. Refer to specifications of the slave device for data addresses and their contents.

Query

The query message specifies the register reference to be preset. The requested preset value is specified as 16-bit data in the query data field.

Here is an example of a request to preset register 1000 to 40150 in slave device 3: (Note that the Starting Address is of 149 or 0x95, less than the force coil 150 by 1.)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x03	0 3	0x03
Function	0x06	0 6	0x06
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x95	9 5	0x95
No. of Registers Hi	0x03	0 3	0x03
No. of Registers Lo	0xE8	E 8	0xE8
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 15. Preset Single Register – Query

Response

The normal response is an echo of the query. Figure 16 shows an example of a response to the query shown in Figure 15.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x03	0 3	0x03
Function	0x06	0 6	0x06
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x95	9 5	0x95
No. of Registers Hi	0x03	0 3	0x03
No. of Registers Lo	0xE8	E 8	0xE8
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 16. Preset Single Register – Response

3.7 Diagnostics (08)

Description

Provides a series of tests for checking the communication system between the master and slave, or for checking various internal error conditions within the slave. Broadcast is not supported.

The function uses a two-byte subfunction code field in the query to define the type of test to be performed. The slave echoes both the function code and subfunction code in a normal response.

Most of the diagnostic queries use a two-byte data field to send diagnostic data or control information to the slave.

For detailed information on Diagnostics Subfunctions, refer to Section 4. Diagnostics Subfunctions (08).

Query

Here is an example of a request to slave device 5 to Return Query Data. This uses a subfunction code of zero (0x0000).

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x08	0 8	0x08
Subfunction Hi	0x00	0 0	0x00
Subfunction Lo	0x00	0 0	0x00
Data Hi	0x03	0 3	0x03
Data Lo	0xE8	E 8	0xE8
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 17. Diagnostics – Query

Response

The normal response to the Return Query Data request is a loopback of the same data. Figure 18 shows an example of a response to the query shown in Figure 17.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x08	0 8	0x08
Subfunction Hi	0x00	0 0	0x00
Subfunction Lo	0x00	0 0	0x00
Data Hi	0x03	0 3	0x03
Data Lo	0xE8	E 8	0xE8
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 18. Diagnostics – Response

3.8 Fetch Communication Event Counter (11, 0x0B)

Description

Returns a status word and an event count from the slave's communication event counter. By fetching the current count before and after a series of messages, a master can determine whether the messages were handled normally by the slave. Broadcast is not supported.

The controller's event counter is incremented once for each successful message completion. It is not incremented for exception responses, poll commands, or fetch event counter commands.

The event counter can be reset by means of the Diagnostics function (08), with a subfunction of Restart Communications Option (code 0x0001) or Clear Counters and Diagnostic Register (code 0x000A).

Query

Here is an example of a request to fetch the communications event counter in slave device 5:

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x0B	0 B	0x0B
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	9	4

Figure 19. Fetch Communications Event Counter – Query

Response

The normal response contains a two-byte status word, and a two-byte event count. Figure 20 shows an example of a response to the query shown in Figure 19.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x0B	0 B	0x0B
Status Hi	0x00	0 0	0x00
Status Lo	0x00	0 0	0x00
Event Count Hi	0x03	0 3	0x03
Event Count Lo	0xE8	E 8	0xE8
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 20. Fetch Communications Event Counter – Response

3.9 Fetch Communication Event Log (12, 0x0C)

Description

Returns a status word and an event count, a message count, and a field of event bytes from the slave. Broadcast is not supported.

The message counter contains the quantity of messages processed by the slave since its last restart, clear counters operation, or power-up.

The message counter can be reset by means of the Diagnostics function (08), with a subfunction of Restart Communications Option (code 0x0001) or Clear Counters and Diagnostic Register (code 0x000A).

Query

Here is an example of a request to fetch the communications event log in slave device 5:

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x0C	0 C	0x0C
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	9	4

Figure 21. Fetch Communications Event Log – Query

Response

The normal response contains a two-byte status word, and 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.

Figure 22 shows an example of a response to the query shown in Figure 21.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x0C	0 C	0x0C
Byte Count	0x08	0 8	0x08
Status Hi	0x00	0 0	0x00
Status Lo	0x00	0 0	0x00
Event Count Hi	0x01	0 1	0x01
Event Count Lo	0xE8	E 8	0xE8
Message Count Hi	0x01	0 1	0x01
Message Count Lo	0xF6	F 6	0xF6
Event 0	0x20	2 0	0x20
Event 1	0x00	0 0	0x00
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	25	12

Figure 22. Fetch Communications Event Log – Response

3.10 Force Multiple Coils (15, 0x0F)

Description

Forces each coil in a sequence of the discrete outputs (DO) to either ON or OFF. When broadcast, the function forces the same coil references in all attached slaves. Refer to specifications of the slave device for data addresses and their contents.

Query

The query message specifies the coil references (the starting coil and the state) to be forced. The requested ON/OFF states are specified by contents of the query data field. Refer to the Read Coil Status (01) for detailed description on how the data field is organized.

The following example shows a request to force a series of 10 coils starting at coil 20 in slave device 5. The binary bits corresponds to the coils in the following way:

Bit	1	1	0	1	0	0	0	1	0	0	0	0	0	1	0	1
Coil	27	26	25	24	23	22	21	20	...	...	...	...	...	30	29	28

Note that the Starting Address is of 19 or 0x13, less than the coil 20 by 1.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x0F	0 F	0x0F
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x13	1 3	0x13
Quantity of Coils Hi	0x00	0 0	0x00
Quantity of Coils Lo	0x0B	0 B	0x0B
Byte Count	0x02	0 2	0x02
Force Data Hi	0xD1	D 1	0xD1
Force Data Lo	0x05	0 5	0x05
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	23	11

Figure 23. Force Multiple Coils – Query

Response

The normal response returns the slave address, function code, starting address, and quantity of coils forced, excluding byte count and force data in the query.

Figure 24 shows an example of a response to the query shown in Figure 23.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x0F	0 F	0x0F
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x13	1 3	0x13
Quantity of Coils Hi	0x00	0 0	0x00
Quantity of Coils Lo	0x0B	0 B	0x0B
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 24. Force Multiple Coils – Response

3.11 Preset Multiple Registers (16, 0x10)

Description

Presets values into a sequence of holding registers. When broadcast, the function presets the same register reference in all attached slaves. Refer to specifications of the slave device for data addresses and their contents.

Query

The query message specifies the register references (the starting register and the data) to be preset. The requested preset values are specified in the query data field.

Here is an example of a request to preset registers 40020 – 40022 in slave device 5 to the following data:

40020 data 0x0164
40021 data 0x0165
40022 data 0x0166

Note that the Starting Address is of 19 or 0x13, less than the register 40020 by 40001.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x10	1 0	0x10
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x13	1 3	0x13
No. of Registers Hi	0x00	0 0	0x00
No. of Registers Lo	0x03	0 3	0x03
Byte Count	0x06	0 6	0x06
Data 1 Hi	0x01	0 1	0x01
Data 1 Lo	0x64	6 4	0x64
Data 2 Hi	0x01	0 1	0x01
Data 2 Lo	0x65	6 5	0x65
Data 3 Hi	0x01	0 1	0x01
Data 3 Lo	0x66	6 6	0x66
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	31	15

Figure 25. Preset Multiple Registers – Query

Response

The normal response returns the slave address, function code, starting address, and quantity of registers preset, excluding byte count and preset data in the query. Figure 26 shows an example of a response to the query shown in Figure 25.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x10	1 0	0x10
Starting Address Hi	0x00	0 0	0x00
Starting Address Lo	0x13	1 3	0x13
No. of Registers Hi	0x00	0 0	0x00
No. of Registers Lo	0x03	0 3	0x03
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 26. Preset Multiple Registers – Response

3.12 Report Slave ID (17, 0x11)

Description

Returns a description of the type of controller present at the slave address, the current status of the slave RUN indicator, and other information specific to the slave device. Broadcast is not supported. The data contents are specific to each type of controller.

Query

Here is an example of a request to report the ID and status of slave device 5:

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x11	1 1	0x11
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	9	4

Figure 27. Report Slave ID – Query

Response

A typical example of a normal response, with slave ID, RUN indicator status, and other device specific data, is shown below.

Figure 28 shows an example of a response to the query shown in Figure 27.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x05	0 5	0x05
Function	0x11	1 1	0x11
Byte Count	Device Specific	Device Specific	Device Specific
Slave ID	Device Specific	Device Specific	Device Specific
RUN Indicator Status	0xFF	F F	0xFF
Additional Data	Device Specific	Device Specific	Device Specific
:	:	:	:
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	Device Specific	Device Specific

Figure 28. Report Slave ID – Response

4. Diagnostic Subfunctions (08)

4.1 Return Query Data (00)

The data passed in the query data field is to be returned (looped back) in the response.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x00	Any 16-bit data	Echo Query Data

4.2 Restart Communications Option (01)

The slave's peripheral port is to be initialized and restarted, and all of its communication event counters are to be cleared. This occurs before the initialization is executed. If the port is currently in Listen Only Mode, no response is returned.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x01	0x00, 0x00 (Exclude event log)	Echo Query Data
0x00, 0x01	0xFF, 0x00 (Initialize all including event log)	Echo Query Data

4.3 Return Diagnostics Register (02)

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x02	0x00, 0x00	Diagnostic Register Contents

4.4 Force Listen Only Mode (04)

Forces the addressed slave to its Listen Only Mode for Modbus communications. This isolates it from the other devices on the network, allowing them to continue communicating without action or response from the addressed slave.

The only function that will be processed after the mode is entered will be the Restart Communications Option function (subfunction 1).

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x04	0x00, 0x00	No Response Returned

4.5 Clear Counters and Diagnostic Register (10, 0x0A)

Clears all counters and the diagnostic register.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x0A	0x00, 0x00	Echo Query Data

4.6 Return Bus Message Count (11, 0x0B)

Returns the quantity of messages that the slave has detected on the communications system.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x0B	0x00, 0x00	Total Message Count

4.7 Return Bus Communication Error Count (12, 0x0C)

Returns the quantity of CRC errors encountered by the slave.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x0C	0x00, 0x00	CRC Error Count

4.8 Return Bus Exception Error Count (13, 0x0D)

Returns the quantity of Modbus exception responses returned by the slave.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x0D	0x00, 0x00	Exception Error Count

4.9 Return Slave Message Count (14, 0x0E)

Returns the quantity of messages addressed to the slave.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x0E	0x00, 0x00	Slave Message Count

4.10 Return Slave No Response Count (15, 0x0F)

Returns the quantity of messages addressed to the slave for which it returned no response.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x0F	0x00, 0x00	Slave No Response Count

4.11 Return Slave Busy Count (17, 0x11)

Returns the quantity of messages addressed to the slave for which it returned a Slave Device Busy exception response.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x11	0x00, 0x00	Slave Device Busy Count

4.12 Return Bus Character Overrun Count (18, 0x12)

Returns the quantity of messages addressed to the slave that it could not handle due to a character overrun condition.

Subfunction	Data Field (Query)	Data Field (Response)
0x00, 0x12	0x00, 0x00	Slave Character Overrun Count

5. Exception Responses

Except for broadcast messages, when a master device sends a query to a slave device it expects a normal response. One of four possible events can occur from the master's query:

- If the slave device receives the query without a communication error, and can handle the query normally, it returns a normal response.
- If the slave does not receive the query due to a communications error, no response is returned. The master program will eventually process a timeout condition for the query.
- If the slave receives the query, but detects a communication error (parity, LRC, or CRC), no response is returned. The master program wil eventually process a timeout condition for the query.
- If the slave receives the query without a communication error, but cannot handle it (for example, if the request is to read a non-existent coil or register), the slave will return an exception response informing the master of the nature of the error.

The exception response message contains Slave Address, Function Code Field, and Data Field.

In an exception response, the slave echoes the slave address in the query, and sets the most-significant bit of the function code to 1. This makes the master's application program recognize the exception response and examine the data field for the exception code.

Figure 29 shows an example of a master query.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x07	0 7	0x07
Function	0x04	0 4	0x04
Starting Address Hi	0x01	0 1	0x01
Starting Address Lo	0x2C	2 C	0x2C
No. of Registers Hi	0x00	0 0	0x00
No. of Registers Lo	0x03	0 3	0x03
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

Figure 29. Read Input Register – Query

If the input register 30301 does not exist, the slave returns an exception response as shown in Figure 30.

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0x07	0 7	0x07
Function	0x84	8 4	0x84
Exception Code	0x02	0 2	0x02
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	11	5

Figure 30. Slave Exception Response

• **Exception Codes:**

Code	Name	Meaning
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the slave.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the slave.
03	ILLEGAL DATA VALUE	A value contained in the query data fields is not an allowable value for the slave.

Appendix A. R1M Series Remote I/O Modbus Communications

The following explanations are applied to typical R1M Series modules as an example. For those models which are not included, please refer to the respective model's data sheet.

The model R1M supports only the RTU framing. ASCII mode is not available.

A-1 Function Codes

The following list shows the function codes supported by the R1M.

Code	Name*	Notes
01 (0x01)	Read Coil Status	Reading DO
02 (0x02)	Read Input Status	Reading DI
03 (0x03)	Read Holding Register	
04 (0x04)	Read Input Register	
05 (0x05)	Force Single Coil	Writing single DO
06 (0x06)	Preset Single Register	
15 (0x0F)	Force Multiple Coils	Writing multiple DO
16 (0x10)	Force Multiple Registers	
	*Based upon Modbus Protocol Reference Guide PI-MBUS-300	

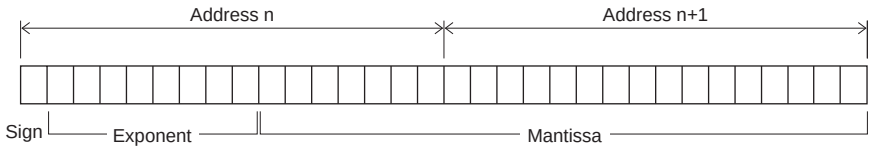
A-2 Data Addresses

	Address	Type				Data	Name
		GH2	J3	A1	D1	Format	
Coil (0X)	1 – 32	---	---	---	Y		DO
	33 – 48	Y	---	---	---		Cold junction compensation SW
Input Status	1 – 32	Y	Y	Y	---		DI
(1X)	33 – 48	Y	Y	---	---		ADC overrange
Input Register	1 – 16	Y	---	---	---	I	AI in %
(3X)	17 – 48	Y	Y	---	---	F	AI per channel in engineering unit
	49 – 50	Y	---	---	---	F	Cold junction temperature
	81 – 96	Y	Y	---	---	I	Channel status
	513	Y	Y	Y	Y	I	System status
	514 – 521	Y	Y	Y	Y	B16	Model No.
	522 – 529	Y	Y	Y	Y	B16	Serial No.
	530 – 537	Y	Y	Y	Y	B16	Hardware version No.
	538 – 545	Y	Y	Y	Y	B16	Firmware version No.
Holding Register	1 – 16	---	---	---	---	I	(Reserved for AO in %)
(4X)	17 – 48	---	---	---	---	F	(Reserved for AO in engineering unit)
	145 – 160	Y	Y	---	---	I	I/O type No.
	161 – 176	Y	---	---	---	I	Burnout type

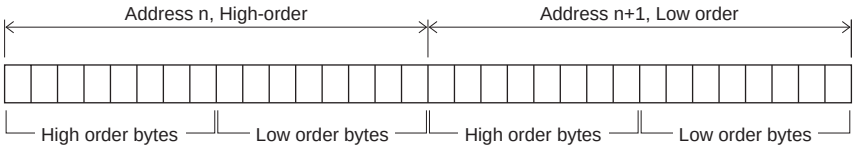
I = 16-bit integer, F = 32-bit floating, B16 = 16-byte character

A-3 Input Data

• 32-bit Floating



• 32-bit Integer, No sign (R1M-A1)



A-4 Coils (DO) Description

DO (1 – 32)

32-point discrete outputs (DO), available only with model R1M-D1.

Cold Junction Compensation SW (33 – 48)

Enabling or disabling the cold junction compensation. 1 = Enable. Available only with thermocouple input type.

A-5 Input Status (DI) Description

DI (10001 – 10032)

32-point discrete inputs (DI), available with models R1M-A1, GH2, and J3.

For GH2 and J3 types, only the address 10001 is available, for use as triggering SW input.

ADC Overrange (10033 – 10048)

Indicating the designated analog input is above or below the full-scale range (0x0000 or 0xFFFF). Available with models R1M-GH2 and J3.

A-6 Input Registers Description

Analog Input in % (30001 – 30016)

Indicating analog input values in percentage for each channel (1 – 16).

Input Type & Range	A/D Data (decimal)
±20V	±20000
±5V	±5000
±1V	±10000
±0.8V	±8000
±0.2V	±20000
±50mV	±5000
±10mV	±10000
Thermocouple	Temperature x 10

Analog Input in Engineering Unit (30017 – 30048)

Indicating analog input values in engineering unit for each channel (1 – 16). The unit is specific to each input type: °C for temperature, Volts for voltage, and % for potentiometer. The data are 32-bit floating values, which requires two consecutive registers for one module.

Cold Junction Temperature (30049 – 30050)

Indicating the current temperature at the cold (reference) junction (°C). Available only for model R1M-GH2. The data is 32-bit floating value.

Channel Status (30081 – 30096)

Indicating the current status of analog inputs (1 – 16 for GH2 type, 1 – 8 for J3 type). The following list shows the names and descriptions of each bit.

Bit	Name	Description
6	Input Overrange	Indicating the designated analog input is in overrange, defined as one or more of the following condition: • ADC input value 0x0000 or 0xFFFF • Burnout status for thermocouple input • Out of range defined in the temperature table (thermocouple and RTD) • ADC error 0 : Normal 1 : Overrange
7	ADC Error	Indicating the status of ADC. 0 : Normal 1 : Error
12	Cold Junction Compensation SW	Indicating whether the cold junction compensation is enabled or disabled, for thermocouple input. 0 : Disable 1 : Enable
Others	Reserved	For system's use

System Status (30513)

Indicating the current system status of the R1M module. The data is 16-bit integer value. The following list shows the names and descriptions of each bit.

Bit	Name	Description
0 – 3	Reserved	For system's use
4	Reserved	For system's use
5	Reserved	For system's use
6	E^2PROM Diagnostics	Indicating the result of sum check for the E^2PROM. 0 : Normal 1 : Sum Check Error
7	ADC Error	Indicating the status of ADC. 0 : Normal 1 : Error
8 – 15	Reserved	For system's use

Model No. (30514 – 30521)

Indicating Model No. of the R1M module. The data is composed of 16-byte characters.

Serial No. (30522 – 30529)

Indicating Serial No. of the R1M module. The data is composed of 16-byte characters.

Hardware Version No. (30530 – 30537)

Indicating Hardware Version No. of the R1M module. The data is composed of 16-byte characters.

Firmware Version No. (30538 – 30545)

Indicating Firmware Version No. of the R1M module. The data is composed of 16-byte characters.

A-7 Holding Registers Description

Analog Output in % (40001 – 40016)

Indicating analog output values in percentage for each channel (1 – 16). Reserved for future use.

Analog Output in Engineering Unit (40017 – 40048)

Indicating analog output values in engineering unit for each channel (1 – 16). The unit is specific to each output type. The data are 32-bit floating values, which requires two consecutive registers for one module. Reserved for future use.

I/O Type No. (40145 – 40160)

Indicating I/O type for each channel (1 – 16). The data are 16-bit integer values.

Model	I/O	I/O Type	Selection	Usable Range	Notes
R1M-GH2	DC input	0x00	-20 – 20 V	-22.7 – 22.7 V	ATT SW ON
		0x01	-5 – 5 V	-5.6 – 5.6 V	ATT SW ON
		0x02	-1 – 1 V	-1.4 – 1.4 V	ATT SW ON
		0x03	-800 – 800 mV	-860 – 860 mV	
		0x04	-200 – 200 mV	-215 – 215 mV	
		0x05	-50 – 50 mV	-53 – 53 mV	
		0x06	-10 – 10 mV	-13.4 – 13.4 mV	
	T/C input	0x10	(PR)	0 – 1760 °C	
		0x11	K (CA)	-270 – 1370 °C	
		0x12	E (CRC)	-270 – 1000 °C	
		0x13	J (IC)	-210 – 1200 °C	
		0x14	T (CC)	-270 – 400 °C	
		0x15	B (RH)	100 – 1820 °C	
		0x16	R	-50 – 1760 °C	
		0x17	S	-50 – 1760 °C	
		0x18	C (WRe 5-26)	0 – 2320 °C	
		0x19	N	-270 – 1300 °C	
		0x1A	U	-200 – 600 °C	
		0x1B	L	-200 – 900 °C	
		0x1C	P (Platinel II)	0 – 1395 °C	
R1M-J3	RTD input	0x30	JPt 100 (JIS '89)	-200 – 500 °C	
		0x31	Pt 100 (JIS '89)	-200 – 660 °C	
		0x32	Pt 100 (JIS '97/IEC)	-200 – 850 °C	
		0x33	Pt 50Ω (JIS '81)	-200 – 649 °C	
		0x34	Ni 508.4Ω	-50 – 280 °C	
		0x35	Pt 1000	-200 – 850 °C	
	POT input	0x40	0 – 100 ohms	0 – 100 %	
		0x41	0 – 500 ohms	0 – 100%	
		0x42	0 – 1k ohms	0 – 100 %	
		0x43	0 – 10k ohms	0 – 100 %	
R1M-D1	DO	0x60			
R1M-A1	DI	0x70			

Burnout Type (40161 – 40176)

Designating burnout action. Available for thermocouple input only.

0 : No burnout
1 : Upscale
2 : Downscale

8-point discrete inputs (DI).

Appendix B. R2M Remote I/O Modbus Communications

The following explanations are applied to typical R2M Series modules as an example. For those models which are not included, please refer to the respective model's data sheet.

The model R2M supports only the RTU framing. ASCII mode is not available.

B-1 Function Codes

The following list shows the function codes supported by the R2M.

Code	Name*	Notes
01 (0x01)	Read Coil Status	Reading DO
02 (0x02)	Read Input Status	Reading DI
03 (0x03)	Read Holding Register	
04 (0x04)	Read Input Register	
05 (0x05)	Force Single Coil	Writing single DO
06 (0x06)	Preset Single Register	
15 (0x0F)	Force Multiple Coils	Writing multiple DO
16 (0x10)	Force Multiple Registers	

*Based upon Modbus Protocol Reference Guide PI-MBUS-300

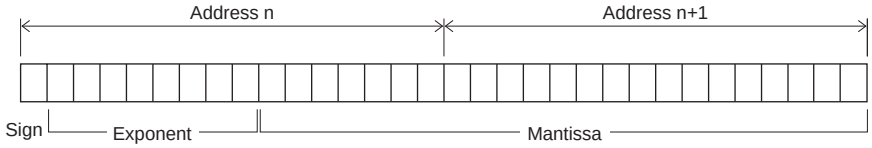
B-2 Data Addresses

	Address	Type G3/H3	Data Format	Name
Coil (0X)	1 – 32	Y		DO
	33 – 40	Y		Cold junction compensation SW
Input Status	1 – 32	Y		DI
(1X)	33 – 40	Y		ADC overrange
Input Register	17 – 32	Y	F	AI per channel in engineering unit
(3X)	49 – 50	Y	F	Cold junction temperature
	81 – 96	Y	I	Channel status
	513	Y	I	System status
	514 – 521	Y	B16	Model No.
	522 – 529	Y	I	Serial No.
	530 – 537	Y	I	Hardware version No.
	538 – 545	Y	B16	Firmware version No.
Holding Register	49 – 50	Y	F	Input filter time constant
(4X)	145 – 152	Y	I	Input type No.
	514	Y	I	Burnout type

I = 16-bit integer, F = 32-bit floating, B16 = 16-byte character

B-3 Input Data

• 32-bit Floating



B-4 Coils (DO) Description

DO (1 – 32)

32-point discrete outputs (DO). Only the address 1 is available, for use as alarm output. Other addresses are reserved for future use.

Cold Junction Compensation SW (33 – 40)

Enabling or disabling the cold junction compensation. 1 = Enable. Available only with thermocouple input type.

B-5 Input Status (DI) Description

DI (10001 – 10032)

32-point discrete inputs (DI).

Only the address 10001 is available, for use as triggering SW input.

ADC Overrange (10033 – 10040)

Indicating the designated analog input is above or below the full-scale range (0x0000 or 0xFFFF).

B-6 Input Registers Description

Analog Input in Engineering Unit (30017 – 30032)

Indicating analog input values in engineering unit for each channel (1 – 8). The unit is specific to each input type: °C for temperature and Volts for voltage. The data are 32-bit floating values, which requires two consecutive registers for one module.

Cold Junction Temperature (30049 – 30050)

Indicating the current temperature at the cold (reference) junction (°C). Available only for thermocouple input type. The data is 32-bit floating value.

Channel Status (30081 – 30088)

Indicating the current status of analog inputs (1 – 8). The following list shows the names and descriptions of each bit.

Bit	Name	Description
6	Input Overrange	Indicating the designated analog input is in overrange, defined as one or more of the following condition: • ADC input value 0x0000 or 0xFFFF • Burnout status for thermocouple input • Out of range defined in the temperature table (thermocouple and RTD) • ADC error 0 : Normal 1 : Overrange
7	ADC Error	Indicating the status of ADC. 0 : Normal 1 : Error
12	Cold Junction Compensation SW	Indicating whether the cold junction compensation is enabled or disabled, for thermocouple input. 0 : Disable 1 : Enable
Others	Reserved	For system's use

System Status (30513)

Indicating the current system status of the R2M module. The data is 16-bit integer value. The following list shows the names and descriptions of each bit.

Bit	Name	Description
0 – 3	Reserved	For system's use
4	Reserved	For system's use
5	Reserved	For system's use
6	E^2PROM Diagnostics	Indicating the result of sum check for the E^2PROM. 0 : Normal 1 : Sum Check Error
7	ADC Error	Indicating the status of ADC. 0 : Normal 1 : Error
8 – 15	Reserved	For system's use

Model No. (30514 – 30521)

Indicating Model No. of the R2M module. The data is composed of 16-byte characters.

Serial No. (30522 – 30529)

Indicating Serial No. of the R2M module. The data is composed of 16-byte characters.

Hardware Version No. (30530 – 30537)

Indicating Hardware Version No. of the R2M module. The data is composed of 16-byte characters.

Firmware Version No. (30538 – 30545)

Indicating Firmware Version No. of the R2M module. The data is composed of 16-byte characters.

B-7 Holding Registers Description

Input Filter Time Constant (40049 – 40050)

Indicating filtering time constant common to all input channels. Effective when high noise is present in input signals. The data are 32-bit floating values, in Seconds. No filtering is performed when the time constant is set to zero (0) second.

Input Type No. (40145 – 40152)

Indicating input type for each channel (1 – 8). The data are 16-bit integer values.

Model	I/O	I/O Type	Selection	Usable Range	Notes
R2M-2G3	DC input	0x00	-10 – 10 V	-10 – 10 V	
R2M-2H3	T/C input	0x10	(PR)	0 – 1760 °C	
		0x11	K (CA)	-270 – 1370 °C	
		0x12	E (CRC)	-270 – 1000 °C	
		0x13	J (IC)	-210 – 1200 °C	
		0x14	T (CC)	-270 – 400 °C	
		0x15	B (RH)	100 – 1820 °C	
		0x16	R	-50 – 1760 °C	
		0x17	S	-50 – 1760 °C	
		0x18	C (WRe 5-26)	0 – 2320 °C	
		0x19	N	-270 – 1300 °C	
		0x1A	U	-200 – 600 °C	
		0x1B	L	-200 – 900 °C	
		0x1C	P (Platinel II)	0 – 1395 °C	

Burnout Type (40514)

Designating burnout action. Available for thermocouple input only.

0 : No burnout
1 : Upscale
2 : Downscale

Appendix C. Modbus TCP/IP Protocol

General descriptions on Modbus TCP/IP protocol are explained in the following sections. For more information, please refer to 'Open Modbus TCP/IP Specification' released on the internet.

C-1 Introduction

Modbus TCP/IP protocol is an expansion of the widely used Modbus protocol into TCP/IP network, which enables Modbus Request/Response transaction on the internet.

In Modbus TCP/IP using 'Client/Server' model for communication, Modbus Master is replaced by Client, Slave by Server. Multiple client/servers are supported, i.e. multiple masters/slaves.

One of the well-known ports, No. 502 is used.

C-2 Protocol Layout

This section describes the encapsulation of a Modbus request or response when it is carried on a Modbus TCP/IP network.

Function and Data fields of Modbus TCP have identical contents to those of Modbus ASCII or RTU mode.

Address field is interpreted differently. Modbus Slave Address field is replaced by a single-byte Unit Identifier which is used to communicate via devices such as bridges, routers and gateways (e.g. model 72EM) that use a single IP address to support multiple independent Modbus end units. For a single Modbus end unit, the IP address is used to identify the slave device, regardless of the contents of the address field.

A dedicated 6-byte long header is added at the head of a Modbus Request/Response message.

Byte 0 : Transaction Identifier (recopied by the server from the received request; typically filled with 0)

Byte 1 : Transaction Identifier (recopied by the server from the received request; typically filled with 0)

Byte 2 : Protocol Identifier (= 0)

Byte 3 : Protocol Identifier (= 0)

Byte 4 : Length (upper digit, = 0 [max. length 256])

Byte 5 : Length (lower digit, number of following bytes)

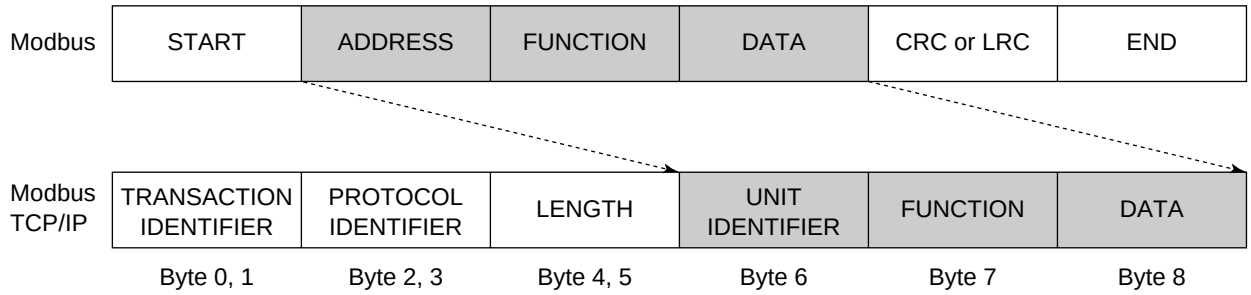
Byte 6 : Unit Identifier (Modbus Slave Address)

Byte 7 : Modbus Function Code

Byte 8 and following bytes : Data Field

The Modbus server copies in the response the transaction identifier of the request. It is typically 0.

A Modbus TCP/IP frame is compared to a general Modbus frame in the figure below.



C-3 Example

Request : Read Holding Register offset by 4 addresses from Unit Identifier 9

Response : 5

Request : 00 00 00 00 00 06 09 03 00 04 00 01

Response : 00 00 00 00 00 05 03 02 00 05

C-4 Point of Caution

LRC or CRC-16 error check field is not required for Modbus TCP/IP. Corruption to a request/response message is detected using TCP/IP or link layer's check mechanisms.

M-SYSTEM WARRANTY

1. What is covered.

M-System Co., Ltd. ("M-System") warrants, only to the original purchaser of new M-System products purchased directly from M-System, or from M-System's authorized distributors or resellers, for its own use not for resale, that the M-System products shall be free from defects in materials and workmanship and shall conform to the specifications set forth in the product catalogue applicable to the M-System products for the Warranty Period (see Paragraph 5 below for the Warranty Period of each product).

THE ABOVE WARRANTY IS THE ONLY WARRANTY APPLICABLE TO THE M-SYSTEM PRODUCTS AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ALL IMPLIED WARRANTIES OF MERCHANTABILITY OR OF FITNESS FOR A PARTICULAR PURPOSE.

2. What is not covered.

This warranty does not cover any M-System product which has been: (1) modified, altered or subjected to abuse, misuse, negligence or accident; (2) improperly installed or installed in conjunction with any equipment for which it was not designed; or (3) damaged or destroyed by disasters such as fire, flood, lightning or earthquake.

In no event shall M-System be liable for any special, incidental, consequential or other damages, costs or expenses (including, but not limited to, loss of time, loss of profits, inconvenience or loss of use of any equipment).

3. Remedies.

If a defective product is returned to M-System in accordance with the procedures described below, M-System will, at its sole option and expense, either: (1) repair the defective product; (2) replace the defective product; or (3) refund the purchase price for the defective product paid by the purchaser. Except as otherwise provided by applicable state law, these remedies constitute the purchaser's **sole and exclusive** remedies and M-System's sole and exclusive obligation under this warranty.

4. Warranty Procedure.

If the purchaser discovers a failure of the M-System products to conform to the terms of this warranty within the Warranty Period, the purchaser must promptly (and, in any event not more than 30 days after the discovery of such failure) notify the relevant party as described below either by telephone or in writing at the below address to obtain an Authorized Return (AR) number and return the defective product to the relevant party. The designated AR number should be marked on the outside of the return package and on all correspondence related to the defective product. The purchaser shall return, at purchaser's expense, defective products only upon receiving an AR number. In order to avoid processing delays, the purchaser must include: copies of the original purchase order and sales invoice; the purchaser's name, address and phone number; the model and serial numbers of the returned product; and a detailed description of the alleged defect.

5. Warranty Period.

Signal Conditioner:	36 months from the date of purchase.
M-Rester:	12 months from the date of purchase.
Valve Actuator:	18 months from the date of shipment from M-System or 12 months from the date of its installation, whichever comes first.
Other Products:	36 months from the date of purchase.

M-SYSTEM CO., LTD.
5-2-55, Minamitsumori, Nishinari-ku,
Osaka 557-0063 JAPAN
Phone: (06) 6659-8201
Fax: (06) 6659-8510
E-mail: info@m-system.co.jp

M·SYSTEM CO.,LTD.
5-2-55, Minamitsumori, Nishinari-ku, Osaka 557-0063 JAPAN
Tel: +81-6-6659-8201 Fax: +81-6-6659-8510
http://www.m-system.co.jp
E-mail: info@m-system.co.jp