



Conformance Test Specification for Modbus TCP

Version 3.0

**Modbus Organization, Inc.
December 15, 2009**

Table of Contents

1.	Summary	3
2.	Purpose of Conformance Testing.....	3
3.	References.....	3
4.	Conventions	4
5.	Definitions.....	4
6.	Test Preparation	4
7.	List of Public Modbus Function Codes	5
8.	List of Valid Exception Responses	6
9.	Protocol Test	7
	Function Code 1, Read Coils	7
	Function Code 2, Read Discrete Inputs	8
	Function Code 3, Read Multiple Holding Registers.....	8
	Function Code 4, Read Input Registers	8
	Function Code 5, Write Single Coil.....	9
	Function Code 6, Write Single Register	9
	Function Code 8, Diagnostics	9
	Function Code 15, Write Multiple Coils.....	9
	Function Code 16, Write Multiple Registers	10
	Function Code 23, Read/Write Multiple Registers.....	11

1. Summary

The following test specification defines the tests used to declare conformance to the Modbus protocol specifications. This test specification defines the criteria for passing the Conformance Test and being declared conformant to the published protocol specification. Also included are the tests the laboratory will use when testing a Modbus server. At this time, a conformance test consists of a Protocol Test of a Modbus server only. The Protocol Test, acting as a Modbus Client, assesses Modbus conformance by checking the server's supported Modbus Function codes. Additionally it will test for message error handling and message format error handling. If the Server passes each required Criterion, it is declared a Conformant Server for the current test revision. The test setup is based on using Modbus TCP. For Modbus TCP devices, the connection is made over TCP/IP. For Modbus Serial devices, a properly configured Schneider Electric ETG-100 Modbus Ethernet-to-Serial Gateway is used.

2. Purpose of Conformance Testing

The goal of conformance testing is to promote and facilitate consistency of server functionality from multiple vendors. Customers benefit from the many possibilities Modbus provides as a versatile and open protocol only when they can be assured that vendors' devices have implemented the protocol correctly.

The Modbus Protocol conformance test verifies that a server is properly responding to the supported Modbus Function Codes. The format of the response is confirmed, however, the specific data returned by a response is not checked. The location and information in non-data fields are analyzed for conformance. This conformance test revision also tests for other aspects of the specification, including exception codes to known corrupted messages or unsupported Function Codes.

3. References

These documents are available at the Modbus Organization website (www.modbus.org):

PI-MBUS-300 Rev. J – Modicon Modbus Protocol Reference Guide (June 1996)

http://www.modbus.org/docs/PI_MBUS_300.pdf (legacy applications only)

Modbus Protocol Specification V1.1b (December 28, 2006)

http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b.pdf

Modbus over Serial Line Specification and Implementation Guide V1.02 (December 20, 2006)

http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf

Modbus Messaging on TCP/IP Implementation Guide V1.0b (October 24, 2006)

http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf

4. Conventions

Unless otherwise noted, all numbers are in decimal.

In specifying byte values, “XX” denotes that the case-specific value should be inserted.

5. Definitions

DUT – Device Under Test. The device being tested for conformance.

IP v4 Address. Internet Protocol Address. This is the 32-bit address assigned to a device. All IP addresses on a network must be unique.

STP (Shielded Twisted-Pair). Cabling used for Ethernet, where strands of wire are twisted together and are shielded either individually or in pairs.

6. Test Preparation

If the Device Under Test is being tested for Modbus TCP conformance, it must be able to support a valid IPv4 address. Suppliers need to ensure that their devices use standard IPv4 addressing. The IP address and the subnet mask must be configured so that the DUT can communicate with the test client.

The test checks for errors in the transmission or reception of Modbus messages only.

The host client uses 255 as the default value for Unit Identifier. If the DUT requires a specific value, this must be included in the DUT documentation. The Unit Identifier value in the conformance test is used to identify a specific downstream Serial device when using a Modbus TCP to Modbus Serial gateway. The range of acceptable addresses for Modbus Serial devices is 1 to 247.

The device documentation must specify which Function Codes are supported by the device from the list of public Function Codes.

The Modbus conformance test supports four types of data references:

- 0x for coils (also known as discrete outputs)
- 1x for discrete inputs
- 3x for input registers
- 4x for holding registers (also known as output registers)

The supplier must provide the following information, so that the test client matches the DUT characteristics:

- State the number of references for each of these data types supported by the device. The protocol assumes that the first memory location is location 0. Any exceptions to this guideline must be noted.
- State whether any of the 0x or 4x data references are read-only.
- State whether there is a unique limit to the maximum number of references that can be read or written in a single command
- State whether there are any non-contiguous memory blocks. The test assumes, by default, that the entire memory area defined is available for reading or writing (as applicable).

Modbus Serial only. The supplier must specify:

- Support for RS-232 and/or RS-485 and wiring instructions
- Support for RTU and/or ASCII modes
- Speed, data bits, stop bits, parity

The supplier must provide customer-quality user documentation to assist the tester with setup instructions.

7. List of Public Modbus Function Codes

Function Code (in dec)	Subcode	Description	Data Type
1		Read Coils	0x
2		Read Discrete Inputs	1x
3		Read Holding Registers	4x
4		Read Input Register	3x
5		Write Single Coil	0x
6		Write Single Register	4x
7		Read Exception Status	Diagnostics
8	0-18, 20 (5-9, 19, and >20 RESERVED)	Diagnostics	Diagnostics
11		Get Com Event Counter	Diagnostics
12		Get Com Event Log	Diagnostics
15		Write Multiple Coils	0x
16		Write Multiple Registers	4x
17		Report Slave ID	Diagnostics
20		Read File Record	File
21		Write File Record	File
22		Mask Write Register	4x
23		Read/Write Multiple Registers	4x
24		Read FIFO Queue	4x
43	13	CANopen Interface Transport	Other
43	14	Read Device ID	Other

8. List of Valid Exception Responses

Modbus Exception Codes		
Code (hex)	Name	Meaning
01	Illegal Function	The Function Code received in the query is not an allowable action for the server/slave. This may be because the Function Code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server/slave is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	Illegal Data Address	The data address received in the query is not an allowable address for the server/slave. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, the PDU addresses the first register as 0, and the last one as 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 4, then this request will successfully operate (as far as address is concerned at least) on registers 96, 97, 98, 99. If a request is submitted with a starting register address of 96 and a quantity of registers of 5, then this request will fail with Exception Code 0x02 “Illegal Data Address,” since it attempts to operate on registers 96, 97, 98, 99 and 100, and there is no register with address 100.
03	Illegal Data Value	A value contained in the query data field is not an allowable value for server/slave. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the Modbus protocol is unaware of the significance of any particular value of any particular register.
04	Slave Device Failure	An unrecoverable error occurred while the server/slave was attempting to perform the requested action.
05	Acknowledge	Specialized use in conjunction with programming commands. The server/slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the client/master. The client/master can next issue a Poll Program Complete message to determine if processing is completed.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The server/slave is engaged in processing a long-duration program command. The client/master should retransmit the message later when the server/slave is free.
08	Memory Parity Error	Specialized use in conjunction with Function Codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The server/slave attempted to read record file, but detected a parity error in the memory. The client/master can retry the request, but service may be required on the server/slave device.
0A	Gateway Path Unavailable	Specialized use in conjunction with gateways to indicate that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0B	Gateway Target Device Failed to Respond	Specialized use in conjunction with gateways to indicate that no response was obtained from the target device. Usually means that the device is not present on the network.

9. Protocol Test

These tests are performed on a network consisting of the DUT and the Laboratory PC. These are connected using TCP/IP.

For example, in the test for Function Code 1, Read Coils, a request is sent via TCP on port 502 to read coils.

The Modbus request will be of the form:

00 00 00 00 00 06 XX 01 00 00 00 01

Corresponding to:

Transaction Identifier: 00 00

Protocol Identifier: 00 00

Message Length: 00 06

Unit Identifier: XX

Function Code: 01

Coil Offset: 00 00

Number of Coils: 00 01

To pass this test, the device must send a response of the form:

00 00 00 00 00 04 XX 01 01 YY

Corresponding to:

Transaction Identifier: 00 00

Protocol Identifier: 00 00

Message Length: 00 04

Unit Identifier: XX

Function Code: 01

Byte Count: 01

Bit Value: YY

Function Code 1, Read Coils

The supplier provides a list of the supported number of 0x references. Unless otherwise noted, the test assumes that the references start with reference no. “00001” (which is specified as 0 in the protocol). All valid references should supply valid responses. The DUT should be able to support read requests of qty-1 to qty-2000 (unless otherwise noted in the product specification). Any request for responses involving a reference outside the valid range should result in the proper exception response.

The DUT needs to respond positively with data to a valid request for the first reference address, the last reference address, and the largest supported block of valid reference addresses. This Function Code should provide an exception/error message for an invalid request, which could include either an illegal number of coils or an illegal reference address.

Pass Criteria: Device responds with the correct number of references to a valid request and responds with the correct exception code in the event of an invalid request.

Function Code 2, Read Discrete Inputs

The supplier provides a list of the supported number of 1x references. Unless otherwise noted, the test assumes that the references start with reference no. “10001” (which is specified as 0 in the protocol). All valid references should supply valid responses. The DUT should be able to support read requests of qty-1 to qty-2000 (unless otherwise noted in the product specification). Any request for responses from a reference outside the valid range should result in the proper exception response.

The DUT needs to respond positively to a valid request for the first reference address, the last reference address, and the largest supported block of valid reference addresses. This Function Code should provide an exception/error message for an invalid request, which could include either an illegal number of discrete inputs or an illegal reference address.

Pass Criteria: DUT responds with the correctly formatted response and number of references to a valid request and responds with the correct exception code in the event of an invalid request.

Function Code 3, Read Multiple Holding Registers

The supplier provides a list of the supported number of 4x references. Unless otherwise noted, the test assumes that the references start with reference no. “40001” (which is specified as 0 in the protocol). All valid references should supply valid responses. The DUT should be able to support read requests of qty-1 to qty-125 (unless otherwise noted in the product specification). Any request for responses from a reference outside the valid range should result in the proper exception response.

The DUT needs to respond positively to a valid request for the first reference address, the last reference address, and the largest supported block of valid reference addresses. This Function Code should provide an exception/error message for an invalid request, which could include either an illegal number of holding registers or an illegal reference address.

Pass Criteria: DUT responds with the correctly formatted response and number of references to a valid request and responds with the correct exception code in the event of an invalid request.

Function Code 4, Read Input Registers

The supplier provides a list of the supported number of 3x references. Unless otherwise noted, the test assumes that the references start with reference no. “30001” (which is specified as 0 in the protocol). All valid references should supply valid responses. The device should be able to support read requests of qty-1 to qty-125 (unless otherwise noted in the product specification). Any request for responses from a reference outside the valid range should result in the correct exception response.

The DUT needs to respond positively with a valid request for the first reference address, the last reference address, and the largest supported block of valid reference addresses. This Function Code should provide an exception/error message for an invalid request, which could include either an illegal number of input registers or an illegal reference address.

Pass Criteria: DUT responds with the correctly formatted response and number of references to a valid request and responds with the correct exception code in the event of an invalid request.

Function Code 5, Write Single Coil

The supplier provides a list of the supported number of 0x references. Unless otherwise noted, the test assumes that the references start with reference no. “00001” (which is specified as 0 in the protocol). All valid references should supply valid responses. Any request for responses from a reference outside the valid range should result in the correct exception response.

The DUT needs to respond positively to a valid request for the reference address, the last reference address, and any specific valid reference address. This Function Code should provide an exception/error message for an invalid request, which could include an illegal or out-of-range reference address.

Pass Criteria: DUT responds with the correctly formatted response to a valid request and responds with the proper exception code in the event of an invalid request.

Function Code 6, Write Single Register

The supplier provides a list of the supported number of 4x references. Unless otherwise noted, the test assumes that the references start with reference no. “40001” (which is specified as 0 in the protocol). Any valid reference should supply a valid response. Any write command to a reference outside the valid range should result in the proper exception response.

The DUT needs to respond positively with a valid write to the first reference address, the last reference address, and any reference in the supported range of valid reference addresses. This Function Code should provide an exception/error message for an invalid write request, which could include an illegal or out of range reference address.

Pass Criteria: DUT responds with the correctly formatted response to a valid request and responds with the proper exception code in the event of an invalid request.

Function Code 8, Diagnostics

The supplier provides a list of supported diagnostic sub-codes. Any request for diagnostic information using a sub-code is expected to respond with the proper sub-code as an echo. The actual diagnostic information returned is not validated. This Function Code should provide an exception/error message for an unsupported sub-code.

Pass Criteria: DUT responds with the correct sub-code to a valid request and responds with the proper exception code in the event of an invalid or unsupported request.

Function Code 15, Write Multiple Coils

The supplier provides a list of the supported number of 0x references. Unless otherwise noted, the test assumes that the references start with reference no. “00001” (which is specified as 0 in the protocol). All valid references should supply valid responses. The

device should be able to support write requests of qty-1 to qty-1968 (unless otherwise noted in the product specification). Any request for responses from a reference outside the valid range should result in the correct exception response.

The DUT needs to positively respond with a valid request for the first reference address, the last reference address, and the largest supported block of valid reference addresses. This Function Code should provide an exception/error message for an invalid request, which could include either an illegal number of coils or an illegal reference address (memory location).

Pass Criteria: DUT responds with the correct number of references to a valid write command, and responds with the proper exception code in the event of an invalid command.

Function Code 16, Write Multiple Registers

The supplier provides a list of the supported number of 4x references. Unless otherwise noted, the test assumes that the references start with reference no. “40001” (which is specified as 0 in the protocol). All valid references should supply valid responses. The DUT should be able to support write requests of qty-1 to qty-123 (unless otherwise noted in the product specification). Any request to write data to a block which includes a reference outside the valid range should result in the correct exception response.

The DUT needs to respond positively with a valid response to a write to the first reference address, the last reference address, and the largest supported block of valid reference addresses. This Function Code should provide an exception/error message for an invalid request, which could include either an illegal number of holding registers or an illegal reference address (memory location).

Pass Criteria: DUT responds with the correct number of references to a valid write command and responds with the proper exception code in the event of an invalid command.

Function Code 23, Read/Write Multiple Registers

The supplier provides a list of the supported number of 4x references. Unless otherwise noted, the test assumes that the references start with reference no. “40001” (which is specified as 0 in the protocol). All valid references should supply valid responses. The DUT should be able to support read requests of qty-1 to qty-125 and write requests of qty-1 to qty-121 (unless otherwise noted in the product specification). Any request for responses from a reference outside the valid range should result in the correct exception response.

The DUT needs to respond positively with a valid request for the first reference address, the last reference address, and the largest supported block of valid reference addresses. This Function Code should provide an exception/error message for an invalid request, which could include either an illegal number of holding registers or an illegal reference address (memory location).

Pass Criteria: DUT responds with the correct number of references to a valid request and responds with the proper exception code in the event of an invalid command.