The University of Michigan Modbus/TCP Conformance Test Laboratory

Conformance Test Policy for the Modbus/TCP Conformance Test Laboratory

Version 1.0

Prepared for:

SCHNEIDER ELECTRIC

Prepared By:

Warren Strong
Laboratory Manager
James Moyne
Laboratory Director

The University of Michigan
Modbus/TCP Conformance Test Laboratory
Engineering Programs Building
2609 Draper St.
Ann Arbor, MI 48109

Phone: 734.764.4336
Fax: 734.936.0347
wstrong@umich.edu
moyne@umich.edu

27 March 2000

1. Summary

1.1. The following are the testing procedures the Laboratory will follow when testing a Modbus/TCP device. At this time, a complete conformance test consists of a Protocol Test as well as an Interoperability Test. The Protocol Test consists of checking TCP connectivity by sending a `PING` request, checking Modbus function codes 0x03 and 0x10, and the SEMI Object Messaging function, 0x5B or its alternate encoding message if either is supported. The Interoperability Test consists of placing the Device Under Test on the Laboratory's loaded network and checking for proper behavior. If the Device Under Test passes the Pass Criteria, it is a Conformant Device for the current test revision.

2. Purpose of Conformance Testing

2.1. The goal of conformance testing is to promote and facilitate interoperability among devices from multiple vendors. Customers will benefit from the many possibilities of this versatile and open protocol only when they can be assured that devices from any vendor will work together efficiently and without conflict.

2.2. The purpose of this revision is to verify that a device conforms to the most basic Modbus/TCP requirements. These are the minimum requirements for a device to be "Modbus/TCP Compliant." The Laboratory will develop tests for the other aspects of the specifications for later revisions of this document.

3. References

These documents are available through the Laboratory Website [1].

[1] The University of Michigan Modbus/TCP Conformance Test Laboratory Website – http://www.eecs.umich.edu/~modbus

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

[3] Open Modbus/TCP Specification Version 1.0 (29 March 1999)

[4] Object Messaging Specification for the Modbus/TCP Protocol Version 1.0 (6 April 1999)

4. Conventions

4.1. Unless otherwise noted, all numbers are in hexadecimal.

4.2. In specifying byte values, "`XX`" denotes that the case-specific value should be inserted here.

5. Definitions

5.1. BOOTP – BOOTstrap Protocol. A protocol used by devices to obtain their IP address from the server when they power-up. The server has a table of network addresses corresponding to IP addresses that is created by the network administrator.

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

5.3. IP Address – Internet Protocol Address. This is the 32-bit address assigned to a device. All IP addresses on a network are unique.

5.4. PLC – Programmable Logic Controller. A device used for reading and writing to devices, running programs, and otherwise managing a network.

5.5. 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

6.1. The DUT must either have its own power supply or be compatible with the power supplies available at the Laboratory. Check the Laboratory's Network Description (see [1]).

6.2. The DUT must also have an IP address. The method used for assigning this address must be specified in the documentation provided with the DUT. The following is a description of acceptable methods.

6.3. *Methods of Assigning IP Addresses*

6.3.1. The Laboratory has a BOOTP server in the PLC. A table entry for the DUT can be created and the DUT can get its IP address via a BOOTP request at start-up.

6.3.2. The vendor may preset the IP address. In this case, the IP address assigned to the device must be in the documentation provided with the DUT. The preferred IP for use on the Laboratory network for a DUT is 192.168.1.100.

6.3.3. The DUT may have other software or hardware methods for setting the IP address. These must be clearly documented in the DUT documentation.

6.4. If a device has a Unit Identifier other than 0, it must also be specified.

6.5. The DUT documentation must also specify whether or not the DUT supports function code 0x5B.

7. Protocol Test

7.1. These tests are performed on a network consisting of the DUT and the Laboratory PC. These are connected via STP Ethernet to a standard hub. Check [1] for more details.

7.2. *PING Test*

7.2.1. Using the Laboratory Windows NT PC, 10 consecutive 32-byte PING requests are sent to the device. The command is:

```
PING -n 10 192.168.1.100
```

7.2.2. The pings should all be returned, corresponding to an output:

```
Reply from 192.168.1.100: bytes=32 time<10ms TTL=128
```

7.2.3. The time elapsed may vary for devices.

7.2.4. Pass Criteria

7.2.4.1. If all 10 pings are returned as specified, the DUT passes the `PING` test.

7.3. *Function 0x03 Test, Read Multiple Registers*

7.3.1. A request is sent via TCP on port 502 to read multiple registers, in the format specified by [2] and [3] from the Laboratory PC.

7.3.2. The Modbus request will be of the form:

```
00 00 00 00 00 06 XX 03 00 00 00 01
```

7.3.2.1. Corresponding to:

Transaction Identifier: `00 00`
Protocol Identifier: `00 00`
Message Length: `00 06`
Unit Identifier: `XX`
Function Code: `03`
Register Offset: `00 00`
Number of Registers: `00 01`

7.3.3. Pass Criteria

7.3.3.1. The response must of the form:

```
00 00 00 00 00 05 00 03 02 XX XX
```

7.3.3.1.1. Corresponding to:

Transaction Identifier: `00 00`
Protocol Identifier: `00 00`
Message Length: `00 05`
Unit Identifier: `00`
Function Code: `03`

Byte Count: `02`
Register Value: `XX XX`

7.3.3.2. At this time, there are no timing requirements for the response. Devices correctly responding to the request pass, regardless of time elapsed.

7.4. *Function 0x10 Test, Write Multiple Registers*

7.4.1. A request is sent via TCP on port 502 to write multiple registers, in the format specified by [2] and [3] from the Laboratory PC.

7.4.2. The Modbus request will be of the form:
`00 00 00 00 00 09 XX 10 00 00 00 01 02 12 34`

7.4.2.1. Corresponding to:
Transaction Identifier: `00 00`
Protocol Identifier: `00 00`
Message Length: `00 09`
Unit Identifier: `XX`
Function Code: `10`
Register Offset: `00 00`
Number of Registers: `00 01`
Number of Bytes: `02`
Data: `12 34`

<u>7.4.3. Pass Criteria</u>

7.4.3.1. The response must of the form:
`00 00 00 00 00 06 00 10 00 00 00 01`

7.4.3.1.1. Corresponding to:
Transaction Identifier: `00 00`
Protocol Identifier: `00 00`
Message Length: `00 06`
Unit Identifier: `00`
Function Code: `10`
Register Offset: `00 00`
Number of Registers: `00 01`

7.4.3.2. At this time, there are no timing requirements for the response. Devices correctly responding to the request pass, regardless of time elapsed.

7.5. *Function 0x5B Test, Object Messaging*

7.5.1. Note: This test is only performed if the DUT supports function code 0x5B.

7.5.2. A request is sent via TCP on port 502 for object messaging, in the format specified by [2], [3] and [4] from the Laboratory PC.

7.5.3. The request for Class 1, Instance 1, Service Code 1 will be issued. This object and service code may or may not be available on the device, however the device must respond with a object access error exception code or the data requested if the object exists.

7.5.4. The Modbus request will be of the form:
`00 00 00 00 00 0B XX 5B 08 40 00 01 00 01 00 01 00`

7.5.4.1. Corresponding to:
Transaction Identifier: `00 00`
Protocol Identifier: `00 00`
Message Length: `00 0B`
Unit Identifier: `XX`

Function Code: 5B
Fragment Byte Count: 08
Fragment Protocol: 40 (01000000 binary, corresponding to a one-fragment message)
Class ID: 00 01
Instance ID: 00 01
Service Code: 00 01
Service Parameter: 00

7.5.5. Response Types

7.5.5.1. This test will generate service response data or a service error code.

7.5.5.2. For a service error code, the response must be of the form:
`00 00 00 00 00 XX 00 5B XX 40 00 01 00 01 00 02 XX ...`

7.5.5.2.1. Corresponding to:
Transaction Identifier: 00 00
Protocol Identifier: 00 00
Message Length: 00 XX
Unit Identifier: 00
Function Code: 5B
Fragment Byte Count: XX
Fragment Protocol: 40 (01000000 binary, corresponding to a one-fragment message)
Class ID: 00 01
Instance ID: 00 01
Service Code: 00 02 (corresponding to the response to service code 00 01)
Service Error Code: XX

7.5.5.3. For service response data, the response must be of the form:
`00 00 00 00 00 XX 00 5B XX XX 00 01 00 01 00 02 00 XX ...`

7.5.5.3.1. Corresponding to:
Transaction Identifier: 00 00
Protocol Identifier: 00 00
Message Length: 00 XX
Unit Identifier: 00
Function Code: 5B
Fragment Byte Count: XX
Fragment Protocol: XX (varies depending on length of message)
Class ID: 00 01
Instance ID: 00 01
Service Code: 00 02 (corresponding to the response to service code 00 01)
Service Parameter Error Code: 00
Service Parameters: XX ...

7.5.6. Pass Criteria

7.5.6.1. If the DUT properly responds with a proper object messaging response for either a service error or service data for the object as defined in 7.5.5.2 and 7.5.5.3, it passes this test.

7.6. *Function 0x5B Test, Object Messaging Alternate Encoding*

7.6.1. Note: This test is performed only if the DUT supports the alternate-encoding method of implementing function code 0x5B.

7.6.2. Note: This test cannot be performed if the device does not pass the Function 0x03 Test, Read Multiple Registers, see 7.3.

7.6.3. Multiple requests are sent via TCP on port 502 to read multiple registers, in the format specified by [2], [3] and [4] from the Laboratory PC.

7.6.4. As specified in Appendix A of [4], function 0x03 requests will be sent, consecutively reading the entire contents of the DUT register memory in 125 register blocks until the ASCII values "SE", "MI" and their zero-sum checksum are found.

7.6.5. The Modbus request will be of the form:
`00 00 00 00 00 06 XX 03 00 00 00 XX`

7.6.5.1. Corresponding to:
Transaction Identifier: `00 00`
Protocol Identifier: `00 00`
Message Length: `00 06`
Unit Identifier: `XX`
Function Code: `03`
Register Offset: `00 00`
Number of Registers: `00 XX` (the best size for reading register blocks will be used here)

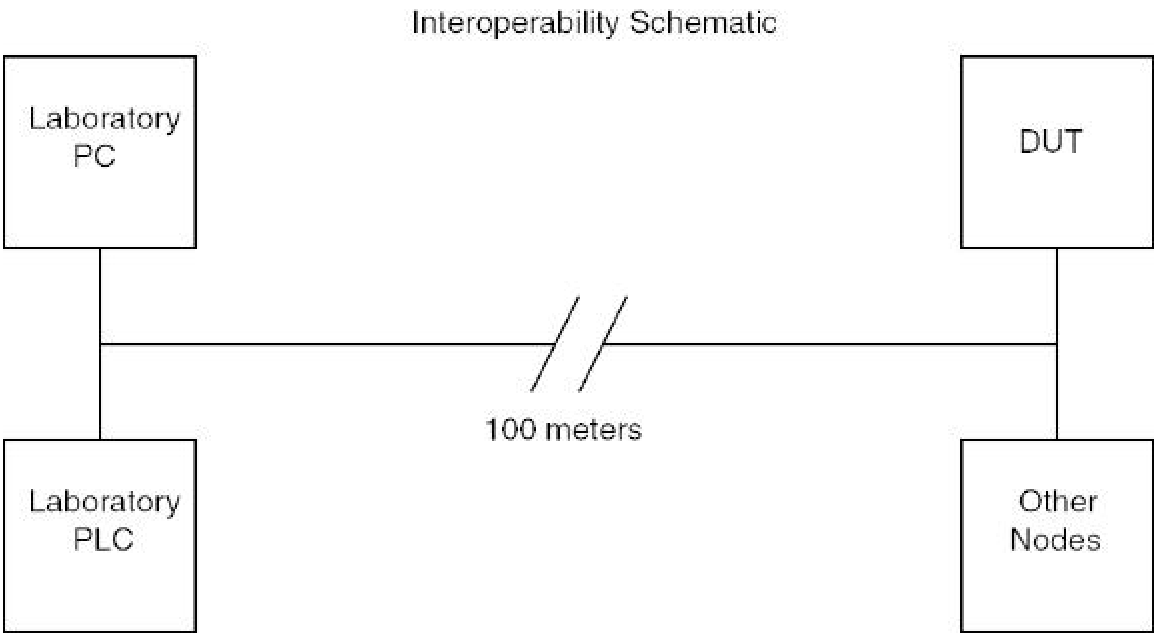
7.6.6. Pass Criteria

7.6.6.1. The registers of data in the response will be checked for the values 0x5345, 0x4D49, 0x5F72 consecutively. These correspond to the ASCII values "SE", "MI" and their zero-sum checksum.

7.6.6.2. If these values are properly returned, the DUT will pass the test.

8. Interoperability Test

8.1. The DUT will be connected to the Laboratory's interoperability network. The description of this network is available on [1]. The DUT and other network slaves are then separated from the Laboratory PC and PLC by 100m of STP Ethernet cable.



8.2. *Schematic of Interoperability Test*

Figure 1: Schematic of Interoperability Test

8.3. With the DUT correctly configured and active on the network, the PLC and PC will generate network traffic to the nodes on the network. This will consist of read/write requests issued very quickly, simulating a large network at with significant network traffic.

8.4. Pass Criteria

8.4.1. The DUT must operate with no detected failures anywhere on the network (including the DUT) for one hour to pass this test.