Introduction to MODBUS

Technical Tutorial

2002 – 12 - 06

Table of Contents

1. Introduction

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

2. Communication between MODBUS devices

MODBUS devices communicate using a master-slave technique in which only one device (the master) can initiate transactions (called queries). The other devices (slaves) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device (I/O transducer, valve, network drive, or other measuring device), which processes information and sends its output to the master using MODBUS.
Masters can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually, but do not respond to broadcast queries

3. MODBUS REGISTER MAP

MODBUS devices usually include a Register Map. MODBUS functions operate on register map registers to monitor, configure, and control module I/O. You should refer to the register map for your device to gain a better understanding of its operation.

4. Serial Transmission Modes of MODBUS networks

The transmission mode defines the bit contents of the message bytes transmitted along the network, and how the message information is to be packed into the message stream and decoded.

Standard MODBUS networks employ one of two types of transmission modes:

4.1. ASCII Mode
4.2. RTU Mode.

The mode of transmission is usually selected along with other serial port communication parameters (baud rate, parity, etc.) as part of the device configuration.

4.1. ASCII Transmission Mode

In the ASCII Transmission Mode (American Standard Code for Information Interchange), each character byte in a message is sent as 2 ASCII characters. This mode allows time intervals of up to a second between characters during transmission without generating errors.

4.2. RTU (Remote Terminal Unit) Transmission Mode

In RTU (Remote Terminal Unit) Mode, each 8-bit message byte contains two 4-bit hexadecimal characters, and the message is transmitted in a continuous stream. The greater effective character density increases throughput over ASCII mode at the same baud rate.

5. MODBUS MESSAGE FRAMING

A message frame is used to mark the beginning and ending point of a message allowing the receiving device to determine which device is being addressed and to know when the message is completed. It also allows partial messages to be detected and errors flagged as a result.

A MODBUS message is placed in a message frame by the transmitting device. Each word of this message (including the frame) is also placed in a data frame that appends a start bit, stop bit, and parity bit.

In ASCII mode, the word size is 7 bits, while in RTU mode; the word size is 8 bits. Thus, every 8 bits of an RTU message is effectively 11 bits when accounting for the start, stop, and parity bits of the data frame

Do not confuse the message frame with the data frame of a single byte (RTU Mode) or 7-bit character (ASCII Mode). The structure of the data frame depends on the transmission mode (ASCII or RTU). Note that on some other network types and on MODBUS Plus, the network protocol handles the framing of messages and uses start and end delimiters specific to the network.

5.1. ASCII Mode Message Frames

ASCII Mode messages start with a colon character ":" (ASCII 3AH) and end with a carriage return-line feed pair of characters (CRLF, ASCII 0DH & 0AH). The only allowable characters for all other fields are hexadecimal 0-9 & A-F. Recall that it only takes 7 significant bits to represent an ASCII character. Likewise, the MODBUS ASCII Mode data 'byte' or character is only 7 bits long.

For ASCII Mode transmission, each character requires 7 data bits. Thus, each character is 10 bits when accounting for the start bit, parity bit, and stop bit of the data frame.

In ASCII Mode, all network devices continuously monitor the network for the 'start of message' colon (:) character. When it is received, every network device decodes the next field to determine if it is the addressed device.

5.2. RTU Mode Message Frames

RTU mode messages start with a silent interval of at least 3.5 character times implemented as a multiple of character times at the baud rate being used on the network. The first field transmitted is the device address. The allowable characters transmitted for all fields are hexadecimal values 0-9, A-F.

A networked device continuously monitors the network, including the silent intervals, and when the first field is received (the address) after a silent interval of at least 3.5 character times, the device decodes it to determine if it is the addressed device. Following the last character transmitted, a similar silent interval of 3.5 character times marks the end of the message and a new message can begin after this interval.

The entire message must be transmitted as a continuous stream. If a silent interval of more than 1.5 character times occurs before completion of the frame (not a continuous stream), the receiving device flushes the incomplete message and assumes the next byte will be the address field of a new message.

In similar fashion, if a new message begins earlier than 3.5 character times following a previous message, the receiving device assumes it is a continuation of the previous message. This will generate an error, as the value in the final CRC field will not be valid for the combined messages.

6. MODBUS ADDRESSES

The master device addresses a specific slave device by placing the 8-bit slave address in the address field of the message (RTU Mode). The address field of the message frame contains two characters (in ASCII mode), or 8 binary bits (in RTU Mode). Valid addresses are from 1-247. When the slave responds, it places its own address in this field of its response to let the master know which slave is responding.

7. MODBUS FUNCTIONS

The function code field of the message frame will contain two characters (in ASCII mode), or 8 binary bits (in RTU Mode) that tell the slave what kind of action to take. Valid function codes are from 1-255, but not all codes will apply to a module and some codes are reserved for future use.

8. MODBUS DATA FIELD

The data field provides the slave with any additional information required by the slave to complete the action specified by the function code. The data is formed from a multiple of character bytes (a pair of ASCII characters in ASCII Mode), or a multiple of two hex digits in RTU mode, in range 00H-FFH. The data field typically includes register addresses; count values, and written data.

If no error occurs, the data field of a response from a slave will return the requested data. If an error occurs, the data field returns an exception code that the master's application software can use to determine the next action to take.

9. MODBUS ERROR CHECKING

MODBUS networks employ two methods of error checking: parity checking

1. Parity checking of the data character frame (even, odd, or no parity)

2. Frame checking within the message frame (Cyclical Redundancy Check in RTU Mode, or Longitudinal Redundancy Check in ASCII Mode).

9.1. Parity Checking

A MODBUS device can be configured for even or odd parity, or for no parity checking. This determines how the parity bit of the character's data frame is set.
If even or odd parity checking is selected, the number of 1 bits in the data portion of each character frame is counted. Each character in RTU mode contains 8 bits. The parity bit will then be set to a 0 or a 1, to result in an even (even parity), or odd (odd parity) total number of 1 bits.

9.2. Frame checking

LRC Longitudinal Redundancy Check (ASCII Mode Only)

In the ASCII transmission mode, the character frame includes an LRC field as the last field preceding the CRLF characters. This field contains two ASCII characters that represent the result of a longitudinal redundancy calculation for all the fields except the starting colon character and ending CR LF pair of characters.

CRC Error Checking (RTU Mode Only)

RTU Mode message frames include an error checking method that is based on a Cyclical Redundancy Check (CRC). The error-checking field of a message frame contains a 16-bit value (two 8-bit bytes) that contains the result of a Cyclical Redundancy Check (CRC) calculation performed on the message contents.

10. MODBUS EXCEPTIONS

If an unsupported function code is sent to a module, then the exception code 01 (Illegal Function) will be returned in the data field of the response message. If a holding register is written with an invalid value, then exception code 03 (Illegal Data Value) will be returned in the response message.

11: MODBUS/TCP

MODBUS/TCP is a communication protocol designed to allow industrial equipment such as Programmable Logic Controllers, computers, operator panels, motors, sensors, and other types of physical input/output devices to communicate over a network.

Modbus/TCP was invented by Modicon/Group Schneider and is today is one of the most popular protocols embedded inside the TCP/IP frames of Ethernet. Modbus/TCP basically embeds a Modbus frame into a TCP frame in a simple manner. This is a connection-oriented transaction, which means every query expects a response.

This query/response technique fits well with the master/slave nature of Modbus, adding to the deterministic advantage that Switched Ethernet offers industrial users. The use of OPEN Modbus within the TCP frame provides a totally scaleable solution from ten nodes to ten thousand nodes without the risk of compromise that other multicast techniques would give.

MODBUS® TCP/IP has became an industry de facto standard because of its openness, simplicity, low cost development, and minimum hardware required to support it.
At this moment there are more than 200 MODBUS® TCP/IP devices available in the market. It is used to exchange information between devices, monitor and program them. It is also used to manage distributed I/Os, being the preferred protocol by the manufacturers of this type of devices.

MODBUS TCP/IP uses TCP/IP and Ethernet to carry the MODBUS messaging structure. MODBUS/TCP requires a license but all specifications are public and open so there is no royalty paid for this license. Making use of TCP/IP also offers the use of embedded Web pages to make life even more user friendly! Simply `surf' your plant intranet for the information you need by using your web browser.

11.1. Performance from a MODBUS TCP/IP system

The performance basically depends on the network and the hardware. If you are running MODBUS® TCP/IP over the Internet, you won't get better than typical Internet response times. However, for communicating for debug and maintenance purposes, this may be perfectly adequate and save you from having to catch a plane or go to site on a Sunday morning!

For a high-performance Intranet with high-speed Ethernet switches to guarantee performance, the situation is completely different.

11.2. How can existing MODBUS devices communicate over MODBUS TCP/IP?

MODBUS® TCP/IP is simply MODBUS® protocol with a TCP wrapper. It is therefore extremely simple for existing MODBUS® devices to communicate over MODBUS® TCP/IP. To do this a gateway device is required to convert MODBUS protocol to MODBUS TCP/IP.

11.3. Advantages of MODBUS/TCP

The key advantages of this protocol can be summarized as follows

- It is scalable in complexity. A device, which has only a simple purpose, need only implement one or two message types to be compliant.
- It is highly scalable in scope. A collection of devices using MODBUS/TCP to communicate can range up to 10,000 or more on a single switched Ethernet network.
- It is simple to administer and enhance. There is no need to use complex configuration tools when adding a new station to a Modbus/TCP network.
- There is no vendor-proprietary equipment or software needed. Any computer system or microprocessor with Internet style (TCP/IP) networking can use MODBUS/TCP.
- It is very high performance, limited typically by the ability of the computer operating systems to communicate. Transaction rates of 1000 per second or more are easy to achieve on a single station, and networks can be easily constructed to achieve guaranteed response times in the millisecond range.
- It can be used to communicate with the large installed base of MODBUS devices, using conversion products, which require no configuration.

12. MODBUS/TCP and Sena Products

Setting new standards at factory floor, Sena Technologies introducing IALink100-MODBUS, an industrial device server that enables RS232/422/485 based modbus based serial devices to be connected to Ethernet using industry standard MODBUS/tcp protocol, for industrial/factory automation.

The IALink100-Modbus is designed to meet the requirements for various industrial applications. Using the IALink100 -Modbus, the users can connect various industrial facilities such as PLC, DCS, DDC, RTU which just support Modbus serial protocol with network devices supporting Modbus/TCP protocols, which let users to access and monitor the facilities at the remote site. It can be easily mounted on to a DIN-rail mounting rack, and it supports screw terminal block interface for power-supply and serial interface.

With these merits for a true communication Data Gateway between Ethernet and Modbus, the Sena IALink100-Modbus can be your best choice to integrate Ethernet and Modbus network within your factory for resource sharing and better network integration.

13. Conclusion

MODBUS is an application layer messaging protocol, positioned at level 7 of the OSI model, that provides client/server communication between devices connected on different types of buses or networks. The industry's serial de facto standard since 1979, MODBUS continues to enable millions of automation devices to communicate. Today, support for the simple and elegant structure of MODBUS continues to grow. The Internet community can access MODBUS at a reserved system port 502 on the TCP/IP stack.

MODBUS is used to monitor and program devices; to communicate intelligent devices with sensors and instruments; to monitor field devices using PCs and HMIs; MODBUS is also an ideal protocol for RTU applications where wireless communication is required.