

Modbus RTU Protocol Overview

[DGH Modbus Analog And Digital I/O Products](#)

The Modbus RTU protocol is an open serial protocol that is widely used in today's industrial monitoring and control equipment. This protocol uses an RS-232 or RS-485 serial interface for communications and is supported by almost every commercial SCADA, HMI, OPC Server and data acquisition software program in the marketplace. Thus, making it very easy to integrate Modbus compatible equipment into new or existing monitoring and control applications and have instant software support.



The Modbus RTU protocol uses a Master/Slave technique to communicate between devices. Meaning, any application that utilizes the Modbus RTU protocol will have a Modbus Master and at least one Modbus Slave. A Modbus Master is typically a host supervisory computer running software that will communicate with one or more Modbus Slave devices.

The Modbus slaves are the devices that perform the system parameter measurements and control the on/off devices in the system. In order to perform these tasks, the Master sends messages to a Modbus Slave requesting a specific task to be performed.

The Modbus RTU protocol uses formatted messages to communicate between Masters and Slaves. Messages are initiated by the system Master and sent to a Slave device. The Slave then responds with the requested data, or an acknowledgement that it performed the requested function. All messages to and from the Master contain a two byte (16-bit) CRC (cyclical redundancy check) checksum value for error checking. In the event that an invalid request is transmitted to a Modbus Slave, a one byte (8-bit) error code value is returned providing a reason why the request is faulty.

The Modbus RTU Master messages contain several important pieces of information. Each message begins with the Slave address of the device to receive the message. Next, is the function number (or command) that will be performed. Followed by any data that is required for the specified function to execute properly. The Slave address is a one byte (8-bit) value that with a usable range of values between 0-247. The Modbus function number is also a one byte value and many standardized function numbers have been defined within the protocol. Any data values required to support the requested function will be one or more bytes in length. The Modbus Slave response message will contain the responding Slave address, the function number and any necessary data requested by the master.

The Modbus RTU protocol is an open protocol and can be used royalty free. This allows manufacturers to use the general Modbus protocol structure in their equipment and add customized functions to access features within their hardware. While this can be a great benefit, it may cause compatibility problems with commercial software programs that do not support the newly defined functions.

The Modbus protocol contains a list of functions that are supported by almost all SCADA, HMI and data acquisition programs in the marketplace today. Below is an abbreviated list of protocol functions used to perform most of the reading and controlling of Modbus devices. From the list, "coils" are typically relays or digital outputs, "registers" represent anything from system status to

analog input or output signal values. Each hardware vendor will have a "Register Map" that details each register location and the possible data values for that register.

Abbreviated list of Modbus Functions:

01 - Read Coil Status

02 - Read Input Status

03 - Read Holding Registers

04 - Read Input Registers

05 - Force Single Coil

06 - Preset Single Register

15 - Force Multiple Coils

10 - Preset Multiple Registers

Modbus Exception Codes

Finally, when specifying Modbus hardware or software, take a moment to review the documentation for each item and ensure that both the hardware and software will support the necessary functions to operate properly. It is a rare event that there will be problems but this review process will eliminate the possibility of surprises later.