

Remote I/O on DOS-based Eagle

Remote I/O on DOS-based Eagle

- Introduction
- Modbus slave address
- Protocol details
- Modbus function codes
- Modbus I/O addresses

The DOS-based Eagle system, running the data logging runtime [DOSLOG](#) version 1.10 and newer, implements the Modicon [Modbus RTU slave protocol](#) over serial line. It can be used through a direct RS232 connection, RS232 through modem, and through RS422/RS485.

From SCADA (Supervisory Control And Data Acquisition) software that can work as a Modbus master, you can read the current value of the analog and digital inputs, and control the analog and digital outputs. The SCADA program can therefore use the Eagle system as a remote I/O as long as it is online.

Each Modbus slave on a Modbus network has a slave address, or node number. It will only respond to a frame from the master if the slave number in the frame matches its own slave number. When using RS422/RS485, you can have multiple Eagle systems on a network, each with different slave addresses. On RS232 (using a direct connection, or through modem), only one slave address is used.

The slave address that the Eagle system responds to, is specified by the `node=nnn` command-line parameter to `DOSLOG.EXE`. The default value is 1. On a multidrop network, you have to make the slave address of each Eagle system unique. See [EagleLog DOS runtime](#).

The serial connection runs with 38400 bps, unless modified by a [command-line parameter](#). The serial parameters are 8 databits, 1 stopbit, no parity, and no hardware handshake. When using RS232 mode, RTS and DTR are always enabled while DOSLOG is running. When using RS422/RS485 mode, the RTS or DTR signal is used to control the transmitter.

The Modbus protocol specifies that messages are separated by a silent interval of at least 3.5 character times (3.6 msec at 9600 bps). If a pause longer than this occurs in the middle of a message, the receiver times out and forgets about the message, and is ready to receive a new one.

This short timeout is a problem when communicating through modems, because modems might insert pauses due to built-in data compression and correction etc. In RS232 mode (direct serial or through modems), DOSLOG therefore uses a timeout of 0.5 second. If a communication error occurs, the master should pause for at least 0.5 second to be sure that the Eagle has forgot about the error and is ready to receive a new one.

In RS422/RS485 mode, the timeout is 3.5 character times plus 0-2 msec. From the end of the communication with another slave on the network and until the Eagle can begin receiving a message, the silent interval must be at least 3.5 character times plus 2 msec. The Eagle can start receiving a new message immediately after it has sent a response of a previous message.

DOSLOG supports the following eight Modbus function codes:

01h - Read Coil Status and **02h - Read Input Status**: These two functions read a consecutive range of digital inputs and digital outputs. Internally they are handled identically, so either of them can be used to read either type of digital I/Os.

03h - Read Holding Registers and **04h - Read Input Registers**: These two functions read a consecutive range of analog inputs, analog outputs, and counters on an ME-2019 I/O module. Internally the functions are handled identically, so either of them can be used to read either type of I/Os. The values returned from analog inputs are the "raw" 12-bit values (0-4095), i.e. no physical value conversion is done.

05h - Force Single Coil: This function sets a single digital output.

06h - Preset Single Register: This function sets a single analog output on an ME-2019 I/O module.

0Fh - Force Multiple Coils: This function sets a consecutive range of digital outputs.

10h - Preset Multiple Registers: This function sets a consecutive range of analog outputs on an ME-2019 I/O module. Since an ME-2019 only has two analog outputs, a range of up to two can be used.

DOSLOG doesn't support broadcast messages.

Note that if you are using [DOSLOG in TSR mode](#), and an output is modified by both the foreground program and a Modbus function at the same time, the result is undefined.

The I/O addresses that exist within an Eagle system depends on which I/O modules are installed. There can be up to 16 I/O modules, numbered 1-16, in a system, and each module has up to 48 individual I/Os.

Before using the Modbus functions in an Eagle system, you must configure the I/O modules with the EagleLog PC program. On the "I/O configuration" page of the Properties for an Eagle system, you tell which modules are installed, and on which address. The numbering of the modules is as specified on that page.

Analog inputs on an ME-2019 module that you want to access through Modbus must have a name. Because reading the analog inputs is relatively slow, DOSLOG only reads those inputs that are actually used, i.e. those that have a name. If you read an unnamed analog input from Modbus, the value FFFFh is returned. With the "ME2019 input mode" settings (voltage range and averages) you select the mode of the input. The "Physical value conversion" settings aren't used since you read the "raw" 0-4095 values through Modbus.

The two counters on an ME-2019 module can be read through Modbus function code 03h and 04h. The counters are always enabled, and only the low 16 bit of their values are returned.

Each Eagle I/O module has up to 48 I/Os of various kinds. For the purpose of Modbus I/O addresses, they are numbered from 1 and up, like this:

ME-1608:

- I/O 1-16 are the 16 digital inputs
- I/O 17-24 are the 8 digital outputs

ME-1648:

- I/O 1-16 are the 16 digital inputs
- I/O 17-46 are the 30 digital outputs

ME-2019:

- I/O 1-8 are the 8 digital inputs
- I/O 9-16 are the 8 digital outputs
- I/O 17-32 are the 16 analog voltage inputs
- I/O 33-40 are the 8 analog current inputs
- I/O 41-42 are the 2 analog outputs
- I/O 43-44 are the 2 counters

ME-4800:

- I/O 1-48 are the 48 digital inputs

The Modbus I/O address of a specific signal is calculated like this:

Modbus_IO_address = (module_number * 100) + IO_number Example: In an Eagle system with three I/O modules, the third one being an ME-2019, the Modbus I/O address of the first analog voltage input is $(3 * 100) + 17 = 317$.

Note that the Modbus standard specifies that Modbus I/O addresses are 1-based (the first one is number 1), but the numbers physically transferred in the address field of a Modbus frame are 0-based (the first one is number 0), i.e. the Modbus master subtracts one from the I/O address when building the frame to send. The Modbus I/O addresses described here are the 1-based numbers.