



Modbus RTU (Master)

Overview

Maple Systems' **Silver Series** Operator Interface Terminals (Maple OITs) communicate with any device that uses the Modbus RTU protocol. The OITs use **MODBUS RTU Extend [PDS V3.00]** protocol driver to allow the Maple OIT to act as the master in a single master, (single slave or multiple slave) format. RS485 networking is supported to connect multiple Modbus slave devices to a single Maple OIT.

The MODBUS RTU Extend V3 protocol driver in EZware also has the following features:

- Network support using RS485 5-wire or RS485 3-wire connection
- Ability to use High Word First format (4x) or Low Word First (5x) format for 32-bit data stored in 4x memory location.
- Ability to read/write to individual bits in 4x memory
- Ability to set a turnaround delay that can be used to slow the rate at which the OIT expects a reply from the PLC/controller for data
- Ability to force a Modbus function code 06 (6x) for controllers that do not support function code 16.

Communications Cable

The Maple OIT should be connected to the controller's serial port that supports Modbus (RS-232 or RS-485 can be used).

A list of communications cables offered by Maple Systems as well as cable assembly instructions to assist you in assembling your own communications cable are available on our website at www.maple-systems.com/cables.htm.

WARNING: If your communications cable is not wired exactly as shown in our cable assembly instructions, damage to the OIT or loss of communications can result.

Controller Settings

The Modbus port on the controller must be set to RTU "slave" mode in order to properly communicate with the OIT (master).

Accessible Memory

Register Memory

The following table lists the controller's register memory ranges that the Maple OITs are able to access. Please note that your controller's memory range may be *smaller* or *larger* than that supported by these OITs. The following register memory can be displayed in 16, 32, or 64 bit format on the Maple OIT.

Controller Register	Type	Controller Register Description
300001 - 365535	3x	Input Registers, Read Only
400001 - 465535	4x	Holding / Output Registers (high-word, low-word order)
500001 - 565535	5x*	Same as 4x but does a word swap (for 32-bit word swapping - low word first)
600001 - 665535	6x**	Force OIT to use function code 06 to write to this register.

NOTES:

* The 5x register type is exactly the same as the 4x memory area. Use the 5x memory area when reading/writing to a 32-bit register using the Low Word First format.

** The 6x register type forces the OIT to execute a function code 06 to write a single register. (The OIT default is to use a function code 16 to write multiple registers, even if it is only writing to one register)

Discrete Memory

The following table lists the controller's discrete memory ranges that the Maple OITs are able to access. Please note that your controller's memory range may be *smaller* or *larger* than that supported by these OITs. The following discrete memory is displayable in single-bit format on the Maple OIT.

Controller Bit	Type	Controller Bit Description
00001 - 09999	0x ^{1,2,3}	Discrete Coils / Outputs
10001 - 19999	1x ^{1,2,3}	Discrete Inputs, Read Only
3x_Bit (see note)	3x_Bit	Input Registers, (bit level access)
4x_Bit (see note)	4x_Bit	Holding/Output Registers, (bit level access)

NOTE:

¹The 4x_Bit memory area is used to read/write to individual bits in the 4x memory area. To use this feature, select the 4x_Bit as Device Type for bit-type objects such as Bit Lamps. Under **Device Address**, use the format nnnbb to enter the word memory area, followed by the two-digit bit reference. For example, to target the 3rd bit of 40015, enter "1502" into the Device Address, (nn=15, bb=02). 3x_Bit device type works the same as for the 4x_Bit, except that it is for accessing bits in a 3x input register.

² The EasyBuilder Modbus driver reads a group of 16 bits at a time. Bit groups are 1-16, 17-32, 33-48, 49-64, etc. All bits in the group must be available in the controller for the OIT to read or errors will result.

Examples:

A) If a Bit Lamp is programmed in the OIT that is addressed for bit 00038, then bits 00032 through 00048 must be available and programmed in the controller.

B) If a Bit Lamp in the OIT is addressed as bit 10068, then bits 10065 through 10080 must be available and programmed in the controller.

³ Unlike the read statements for bits, the EasyBuilder Modbus driver will write to just one bit at a time; however, whenever a bit write occurs, the OIT will automatically execute a read. Therefore, even if the intention is to only write to one bit, all sixteen bits must be available and programmed into the PLC that includes the bit being written to.

Address Format when Networking

If you are connecting multiple PLCs/Controllers on a network to an OIT, you can specify the network node address for each object placed onto the OIT screen. To enable multiple node addressing, set the “Address Mode” (found under EDIT=>System Parameters=>{EDITOR}tab) to “Extended”. To target a specific slave address, you must use the following format when entering the address in the Device Address box: aaa#nnnn where aaa=network address (1-255) and nnnn=memory address. The pound sign (#) is used as a placemaker. For example, to configure the Numeric Data object to read memory address 40015 of a PLC that has been assigned a network address of 2: Device Type=4x, Device Address=2#15.

Network Settings

When connecting multiple PLCs/controllers to a single Maple OIT, you must also set the following in the Edit-Set System Parameters menu:

- On the PLC tab, select RS-485 4w for a 4-wire cabling scheme. Select RS-485 2w for a 2-wire cabling scheme (TXD+ and RXD+ connected together, TXD- and RXD- connected together).
- On the Editor tab in System Parameters, select 'Extended' for Address Mode.

The OIT uses the following Modbus codes:

Data Type	Read/Write	Description	Uses Modbus Code
0x	R	Read output bit [read coil]	01
0x	W	Write output bit [write coil]	05
1x	R	Read input bit [read discrete input]	02
3x	R	Read input register [read input register]	04
3x_Bit	R	Read input register - bit level [read input register's bit]	04
4x	R	Read holding register (normal word order)	03
4x	W	Write holding register (normal word order)	16
4x_Bit	R	Read holding register - bit level	03
4x_Bit	W	Write holding register - bit level	16
5x	R	Read holding register (double word, swap word order)	03
5x	W	Write holding register (double word, swap word order)	16
6x	R	Read holding register	03
6x	W	Write holding register (force a function code 06 when writing)	06

EZware Settings

The following table lists the communications settings that must be configured in EZware. These settings can be found in the Edit-Set System Parameters menu under the PLC tab. Please note:

- the **Recommended Settings** column provides the recommended setting based upon default settings most commonly used in Modbus Devices
- the **Options** column lists EZware’s options; your controller may not support every option

Name	Recommended Settings	Options	Important Notes
PLC type:	Modbus RTU Extend [PDS V3.00]		
Serial port I/F:	RS232	RS232, RS485	Must match the controller/PLC
Data Bits:	8	7 or 8	Must match the controller/PLC
Stop Bits:	1	1 or 2	Must match the controller/PLC
Baud Rate:	19200	9600,19200, 38400,57600, 115200	Must match the Controller’s port setting. Use the fastest baud rate supported by the Controller.
Parity:	None	Even, Odd, None	Must match the controller/PLC
HMI station No.:	0	N/A	Does not apply to this protocol.
PLC station No.:	1	0*, 1-255	Set to PLC/Controller node addr.
Multiple HMI:	Disable	Master, Slave	use for multiple OITs
HMI-HMI link speed:	38400	38400, 115200	use for multiple OITs
PLC time out constant (sec)	2.0	1.5 to 5.0	adjust if longer timeout is required
PLC block pack:	0	0-10	see <i>Silver Series Installation and Operation Manual</i>
Parameter 2	15 (typ. When RS485-2W is used)	0-1000	Turn around delay in milliseconds

Note: (*) The OIT does not support broadcast messages. The OIT always expects a reply message after it sends a request. Some Modbus controllers use address 0 as a broadcast message so that all node addresses will listen and process the command, but they will not reply back to the master (such as a broadcast ‘RESET’ command). If, however, the controller has a valid address of 0 and it is not used as a broadcast address, then address 0 can be used.