# Table Of Contents

MODBUS: Help Rev: 2/12/04.....................................................................................................i
Using MODBUS: Unitronics' PLCs, Master - Slave ................................................................. 1
Using MODBUS: Accessing PLC data via SCADA/OPC server ............................................. 2
FB Operations .......................................................................................................................... 3
Examples .................................................................................................................................. 4
MODBUS: Configuration.............................................................................................................. 5
MODBUS: ScanEX and Scan...................................................................................................... 6
Read Coils (1) .............................................................................................................................. 7
Read Inputs (2) ........................................................................................................................... 8
Read Holding Registers (3)... ........................................................................................................ 8
Read Float Registers (3)... ........................................................................................................... 9
Read Input Registers (4)... ......................................................................................................... 10
Read Float Registers (4)... ......................................................................................................... 11
Force Coil (5)... ............................................................................................................................ 12
Preset Holding Register (6)... ..................................................................................................... 13
Loopback Test (8)... ................................................................................................................... 14
Force Coils (15)... ..................................................................................................................... 14
Preset Holding Registers (16)... ............................................................................................... 15
Preset Float Registers (16)... ..................................................................................................... 16
Read/Write from Data Tables ....................................................................................................17
  Read from Data Table ............................................................................................................... 17
  Write to Data Table .................................................................................................................. 18
Configuring a MODBUS slave device........................................................................................ 19
Slave Addressing....................................................................................................................... 20
  Slave Address Tables ............................................................................................................... 20
MODBUS via GSM or Standard Modem .................................................................................... 23
MODBUS Status Operands ........................................................................................................ 24
Index ............................................................................................................................................ 29
MODBUS enables you to establish master-slave communications with any connected device that supports the MODBUS protocol. Any controller in the network may function as either master or slave using any of the controller's existing COM Ports.

Within a MODBUS network, you can use standard MODBUS commands to read and write bit and register data; you can also read and write data to Vision controller Data Tables.

Unitronics currently supports RTU (binary) transmission mode.

Using MODBUS: Unitronics’ PLCs, Master - Slave

Before using a MODBUS operation in your application, you must:

- Synchronize the communication port settings of master and slave devices. This is done by placing COM Port Init FBs, set with identical parameters, in the ladder application of both master and slave.

- Include at least 1 MODBUS Configuration FB in the ladder application of both master and slave. The port you select must be the same port selected in the COM Port Init FB.

- The condition that activates the Configuration must turn ON for a single program scan (positive transition recommended). However, the MODBUS configuration must be scanned during every program cycle--after the Configuration is activated. One way to ensure this is by placing the configuration in the first subroutine of the main module.

- Enable slave devices to be accessed by placing a Scan_EX FB in the slave's Ladder application.

The figure below shows the elements required to carry out a Read Coils Operation.
Note that the operand addresses in slave PLCs are indirect addresses (pointers).

If Slave ID is #95, the master accesses the slave configured as MODBUS ID 95.

If M17 contains the value 5610, the master will begin 'reading' from M10 in the slave.

Here, 32-bit ML registers are written into 16-bit Ml's. Note that only the last (high) 16 bits of an ML will be transferred into an Ml.

This contains the ID# of the device the master will access.

This value 'points' to the start of the vector of registers in the slave.

This value sets the length of the register of vectors to be read in both master and slave.

If the Read Length is #10, the master reads 10 operands. Note that the Read Length cannot exceed the number of operands of that type. For example, since there are only 256 ML registers, 257 is an illegal value.

Using MODBUS: Accessing PLC data via SCADA/OPC server

The PC master can access data within the PLC via the addresses given in the Slave Addresses Table.

The PLC slave's Ladder application must include the following:

- A COM Port Init FB.
- A MODBUS Configuration FB. Within the Configuration, the port you select must be the same port selected in the COM Port Init FB.
  
  **Note** The condition that activates the Configuration must turn ON for a single program scan (positive transition recommended). However, the MODBUS configuration must be scanned during every program cycle--after the Configuration is activated. One way to ensure this is by placing the configuration in the first subroutine of the main module.
- A Scan_EX FB.
Note ♦ The operand addresses in slave PLCs are indirect addresses (pointers).

FB Operations

Operations are grouped under MODBUS on the FB’s menu.
MODBUS

MODBUS: Configuration

MODBUS: Scan

MODBUS: Read Coils (1)
MODBUS: Read Inputs (2)

Read Holding Registers (3)
Read Float Registers (3)
Read Input Registers (4)
Read Float Input Registers (4)

Force Coil (5)
Preset Holding Register (6)
Loopback Test (8)
Force Coils (15)

Preset Holding Registers (16)
Preset Float Registers (16)

Read/Write to Data Tables

Examples

Sample applications may be found in the VisiLogic Examples folder. This folder contains field-tested VisiLogic (.vlp) sample applications. You can open this folder via the Help Menu.
MODBUS

The folder is typically located at: C:\ProgramFiles\Unitronics\VisiLogic\Examples\Verx.xx, where x.xx indicates the version of VisiLogic.

MODBUS: Configuration

A MODBUS Configuration FB must be included in both master and slave Ladder applications as shown below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Number</td>
<td>Constant</td>
<td>Click the drop-down arrows to view available ports; click the port you want to use.</td>
</tr>
<tr>
<td>Network ID</td>
<td>Constant</td>
<td>This number identifies the device on the network. You can either assign an ID via an MI, or directly via a constant number. The unit ID range is from 0-255. Do not assign the same ID number to more than one device.</td>
</tr>
<tr>
<td>Time out</td>
<td>Constant or MI</td>
<td>This is the amount of time a master device will wait for an answer from a slave. Time out units are defined in 10 msecs; a Time out value of 100 is equal to 1 second.</td>
</tr>
<tr>
<td>Retries</td>
<td>Constant or MI</td>
<td>This is the number of times a device will try to send a message.</td>
</tr>
<tr>
<td>Function in Progress</td>
<td>MB</td>
<td>This bit is ON when MODBUS is active. Use this as a condition bit for MODBUS operations to avoid communication conflicts.</td>
</tr>
</tbody>
</table>

Note ♦ Indirectly addressed parameters in a MODBUS Configuration FB are only read when the Configuration is called. Since a Configuration is generally called as a power-up task, if, for example Retries has been indirectly addressed, and the linked MI is updated, the new value will not be read into the Configuration. The value will only be updated until the Configuration is called.

♦ While a master attempts to send a command, the Function In Progress bit is ON. The number of attempts that the master will make is the number in Retries +1, where ‘1’ is the initial access attempt.

♦ When a master attempts to access a slave device, and the slave does not answer, the Function In Progress bit will turn ON. This bit will remain on
according to the following:
(the number of retries + 1) x (Time Out), where '1' is the initial access attempt.
Note that the Time Out parameter is in units of 10 msec.

The Ladder application below enables the controller act as a MODBUS master and read coils in a slave PLC. The Scan_EX operation in the final net enables the controller to also act as a slave.

MODBUS: ScanEX and Scan

Scan_EX enables a master device to access a slave PLC.
Note ♦ Scan_Ex is recommended for new applications.

About Scan and Scan_EX

MODBUS Versions previous to V2.01 offered only the Scan FB. Scan is still supported for older, working applications. When MODBUS operations accessed double registers (5100 addresses and higher), using odd addresses, such as 5101, there were incompatibility issues.

Scan_EX is recommended for new applications.

When ScanEX receives an input parameter in the 32-bit range (for example, 5100{ML}), it automatically takes double-register values.

If, for example, ScanEX receives a Read Register(6) request for 5100, it returns the values in 5100 and 5101. If, however, ScanEX receives Read Register(6) request for 5101, it returns Status Message #2-- since 5101 provides the 'high' bytes of the 32-bit register, it is not a legal address.

Read Coils (1)

Use this command to read the status of a selected group of coils and write them into a vector. The coil's status is written into a vector of MBs in the master PLC.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave ID</td>
<td>Constant or MI</td>
<td>The ID of the slave device containing the coils to be read (data source).</td>
</tr>
<tr>
<td>Slave: Start of Vector</td>
<td>Constant, MI, ML, or DW</td>
<td>The start of the vector of coils to be read (data source). Note ♦ Check topic Slave Address Tables</td>
</tr>
<tr>
<td>Read: Vector Length</td>
<td>Constant or MI</td>
<td>The vector length. Note ♦ A MODBUS command cannot read/write more than 1900 bit operands at one time. In addition, 0 is not a legal length.</td>
</tr>
<tr>
<td>Master: Start of Vector</td>
<td>MB</td>
<td>This is the start of a vector of MBs that will contain the coils' status in the master (data destination).</td>
</tr>
<tr>
<td>Status Messages</td>
<td>MI</td>
<td>Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.</td>
</tr>
<tr>
<td>Total Sessions</td>
<td>DW</td>
<td>This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.</td>
</tr>
</tbody>
</table>
Acknowledgements

This is the number of times the slave device answers.

**Read Inputs (2)**

Use this command to read the status of a selected group of inputs in a slave device and write them into a vector. The inputs' status is written into a vector of MBs in the master PLC.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave ID</td>
<td>Constant or MI</td>
<td>The ID of the slave device containing the inputs to be read (data source).</td>
</tr>
<tr>
<td>Slave: Start of Vector</td>
<td>Constant, MI, ML, or DW</td>
<td>The start of the vector of inputs to be read (data source). Note ♦ Check topic Slave Address Tables</td>
</tr>
<tr>
<td>Read: Vector Length</td>
<td>Constant or MI</td>
<td>The vector length. Note ♦ A MODBUS command cannot read/write more than 1900 bit operands at one time. In addition, 0 is not a legal length.</td>
</tr>
<tr>
<td>Master: Start of Vector</td>
<td>MB</td>
<td>This is the start of a vector of MBs that will contain the inputs' status in the master (data destination).</td>
</tr>
<tr>
<td>Status Messages</td>
<td>MI</td>
<td>Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.</td>
</tr>
<tr>
<td>Total Sessions</td>
<td>DW</td>
<td>This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>DW</td>
<td>This is the number of times the slave device answers.</td>
</tr>
</tbody>
</table>

**Read Holding Registers (3)**

Use this command to read the values of a selected group of registers in a slave PLC and write them into a defined vector of registers in the master.
Parameter | Type          | Function                                                                 |
---         | ---           | ---                                                                        |
Slave ID    | Constant or MI | The ID of the device containing the registers to be read (data source).  |
Slave: Start of Vector | Constant, MI, ML, or DW | The start of the vector of registers to be read (data source). Note ♦ Check topic Slave Address Tables |
Read: Vector Length | Constant, MI, ML, or DW | The vector length Note ♦ A MODBUS command cannot read more than 124 16-bit integers, 62 double registers, or 64 float registers at one time. In addition, 0 is not a legal length. ♦ If, within the Slave: Start of Vector parameter, the selected register type is a 32-bit double register (slave addresses 5100 and greater) the preset vector length must be doubled as well. If, for example:
  - Slave: Start of Vector parameter is set to 6300, and
  - You wish to preset 4 registers, for a total of 16 bytes
  - You must set the Preset Vector length to 8.
  Note that this means that, in these cases, the Preset: Vector Length parameter will always be an even number. |
Master: Start of Vector | MI | This is the start of a vector of MIs that will contain the registers' values in the master (data destination). |
Status Messages | MI | Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages. |
Total Sessions | DW | This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW. |
Acknowledgements | DW | This is the number of times the slave device answers. |

**Read Float Registers (3)**

Use this command to read the values of a selected group of floating point registers in a slave device and write them into a defined vector of registers in the master. Values after the decimal point are rounded to the nearest whole value.
MODBUS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave ID</td>
<td>Constant or MI</td>
<td>The ID of the device containing the registers to be read (data source).</td>
</tr>
<tr>
<td>Slave: Start of Vector</td>
<td>Constant, MI, ML, or DW</td>
<td>The start of the vector of registers to be read (data source). Note ♦ Check topic Slave Address Tables</td>
</tr>
<tr>
<td>Read: Vector Length</td>
<td>Constant, MI, ML, or DW</td>
<td>The vector length Note ♦ A MODBUS command cannot read more than 124 16-bit integers, 62 double registers, or 64 float registers at one time. In addition, 0 is not a legal length. ♦ If, within the Slave: Start of Vector parameter, the selected register type is a 32-bit double register (slave addresses 5100 and greater) the preset vector length must be doubled as well. If, for example: - Slave: Start of Vector parameter is set to 6300, and - You wish to preset 4 registers, for a total of 16 bytes - You must set the Preset Vector length to 8. Note that this means that, in these cases, the Preset: Vector Length parameter will always be an even number.</td>
</tr>
<tr>
<td>Master: Start of Vector</td>
<td>MI</td>
<td>This is the start of a vector of MIs that will contain the registers' values in the master (data destination).</td>
</tr>
<tr>
<td>Status Messages</td>
<td>MI</td>
<td>Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.</td>
</tr>
<tr>
<td>Total Sessions</td>
<td>DW</td>
<td>This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>DW</td>
<td>This is the number of times the slave device answers.</td>
</tr>
</tbody>
</table>

Read Input Registers (4)

Use this command to read the values of a selected group of registers in a slave PLC and write them into a defined vector of registers in the master.
Parameter | Type | Function
--- | --- | ---
Slave ID | Constant or MI | The ID of the device containing the registers to be read (data source).
Slave: Start of Vector | Constant, MI, ML, or DW | The start of the vector of registers to be read (data source). **Note** Check topic Slave Address Tables
Read: Vector Length | Constant, MI, ML, or DW | The vector length  **Note** A MODBUS command cannot read more than 124 16-bit integers, 62 double registers, or 64 float registers at one time. In addition, 0 is not a legal length.  ♦ If, within the Slave: Start of Vector parameter, the selected register type is a 32-bit double register (slave addresses 5100 and greater) the preset vector length must be doubled as well.  ♦ If, for example:  - Slave: Start of Vector parameter is set to 6300, and  - You wish to preset 4 registers, for a total of 16 bytes  - You must set the Preset Vector length to 8.  Note that this means that, in these cases, the Preset: Vector Length parameter will always be an even number.
Master: Start of Vector | MI | This is the start of a vector of MIs that will contain the registers' values in the master (data destination).
Status Messages | MI | Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.
Total Sessions | DW | This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.
Acknowledgements | DW | This is the number of times the slave device answers.

**Read Float Registers (4)**

Use this command to read the values of a selected group of floating point registers in a slave device and write them into a defined vector of registers in the master. Values after the decimal point are rounded to the nearest whole value.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave ID</td>
<td>Constant or MI</td>
<td>The ID of the device containing the registers to be read (data source).</td>
</tr>
</tbody>
</table>
| Slave: Start of Vector     | Constant, MI, ML, or DW | The start of the vector of registers to be read (data source).  

**Note**: Check topic Slave Address Tables

| Read: Vector Length        | Constant, MI, ML, or DW | The vector length  

**Note**: A MODBUS command cannot read more than 124 16-bit integers, 62 double registers, or 64 float registers at one time. In addition, 0 is not a legal length.  
If, within the Slave: Start of Vector parameter, the selected register type is a 32-bit double register (slave addresses 5100 and greater) the preset vector length must be doubled as well. 
If, for example:  
- Slave: Start of Vector parameter is set to 6300, and  
- You wish to preset 4 registers, for a total of 16 bytes  
- You must set the Preset Vector length to 8. 
Note that this means that, in these cases, the Preset: Vector Length parameter will always be an even number.

<table>
<thead>
<tr>
<th>Master: Start of Vector</th>
<th>MI</th>
<th>This is the start of a vector of MIs that will contain the registers' values in the master (data destination).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status Messages</td>
<td>MI</td>
<td>Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.</td>
</tr>
<tr>
<td>Total Sessions</td>
<td>DW</td>
<td>This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>DW</td>
<td>This is the number of times the slave device answers.</td>
</tr>
</tbody>
</table>

**Force Coil (5)**

Use this command to force the status of a selected coil in a slave PLC. The coil’s status is forced according to the status of a selected MB in the master PLC.
### Parameter | Type | Function
--- | --- | ---
Slave ID | Constant or MI | The ID of the device containing the coil to be forced (data source).
Slave Address | Constant, MI, ML, or DW | The address of the coil to be forced (data target). **Note**: Check topic Slave Address Tables
Value to Force | M, SB, I, O, T | This MB is located in the master PLC; this MB contains the status to be forced (data source). If, for example, the status of this MB is OFF, the status of the coil in the slave will be forced to OFF. **Note**: A MODBUS command cannot read/write more than 1900 bit operands at one time. In addition, 0 is not a legal length.
Status Messages | MI | Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.
Total Sessions | DW | This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.
Acknowledgements | DW | This is the number of times the slave device answers.

### Preset Holding Register (6)

Use this command to preset the value of a single register in a slave PLC. The value is set in a register contained in the master PLC.
### MODBUS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave ID</td>
<td>Constant or MI</td>
<td>The ID of the device to be checked.</td>
</tr>
<tr>
<td>Slave: Operand Address</td>
<td>Constant, MI, ML, or DW</td>
<td>The address of the register to be preset (target). Check topic Slave Address Tables</td>
</tr>
<tr>
<td>Value to Preset</td>
<td>Constant, MI, SI, ML, SL, DW, SDW or T</td>
<td>This is the address of the register containing the value in the master PLC (source). This value will be written into the slave's register, the register that is to be preset.</td>
</tr>
<tr>
<td>Status Messages</td>
<td>MI</td>
<td>Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.</td>
</tr>
<tr>
<td>Total Sessions</td>
<td>DW</td>
<td>This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>DW</td>
<td>This is the number of times the slave device answers.</td>
</tr>
</tbody>
</table>

### Loopback Test (8)

Use this command to send a test message to a slave device and receive Acknowledgements when communications are functioning properly.

### Force Coils (15)

Use this command to force the status of a selected group of coils in a slave PLC. The coils’ status is forced according to the status of a group of MBs in the master PLC.
### MODBUS Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave ID</td>
<td>Constant or MI</td>
<td>The ID of the slave device containing the coils to be forced (target).</td>
</tr>
<tr>
<td>Slave: Start of Vector</td>
<td>Constant, MI, ML, or DW</td>
<td>The start of the vector of coils to be forced (data target). Note ♦ Check topic Slave Address Tables</td>
</tr>
<tr>
<td>Master: Start of Vector</td>
<td>MI, SB, I, O, T</td>
<td>This is the start of a vector of MBs that will contain the coils' status in the master (data source).</td>
</tr>
<tr>
<td>Force: Vector Length</td>
<td>Constant or MI</td>
<td>The vector length. Note ♦ A MODBUS command cannot read/write more than 1900 bit operands at one time. In addition, 0 is not a legal length.</td>
</tr>
<tr>
<td>Status Messages</td>
<td>MI</td>
<td>Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.</td>
</tr>
<tr>
<td>Total Sessions</td>
<td>DW</td>
<td>This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>DW</td>
<td>This is the number of times the slave device answers.</td>
</tr>
</tbody>
</table>

### Preset Holding Registers (16)

Use this command to preset the value of a group of registers in a slave PLC. The values are set in a vector of registers contained in the master PLC.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave ID</td>
<td>Constant</td>
<td>The ID of the device containing the registers to be preset (target).</td>
</tr>
</tbody>
</table>
### Preset Float Registers (16)

Use this command to preset the value of a group of floating point registers in a slave PLC. The values are set in a vector of registers contained in the master PLC. Values after the decimal point are rounded to the nearest whole value.

#### Parameter | Type | Function
--- | --- | ---
Slave ID | Constant or MI | The ID of the device containing the register to be preset (target).
Slave: Start of Vector | Constant, MI, ML, or DW | The address of the register to be preset (target).

**Note**: Check topic Slave Address Tables.
Master: Start of Vector
MI, SI, ML, SL, DW, SDW or T
This is the address of the register containing the value in the master PLC (source). This value will be written into the slave's register, the register that is to be preset.

Preset: Vector Length
Constant, MI, ML, or DW
The length of the vector of registers in both master and slave.

Note ♦ A MODBUS command cannot read more than 124 16-bit integers, 62 double registers, or 64 float registers at one time. In addition, 0 is not a legal length.
♦ If, within the Slave: Start of Vector parameter, the selected register type is a 32-bit double register (slave addresses 5100 and greater) the preset vector length must be doubled as well.
If, for example:
- Slave: Start of Vector parameter is set to 6300, and
- You wish to preset 4 registers, for a total of 16 bytes
- You must set the Preset Vector length to 8.
Note that this means that, in these cases, the Preset: Vector Length parameter will always be an even number.

Status Messages
MI
Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.

Total Sessions
DW
This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.

Acknowledgements
DW
This is the number of times the slave device answers.

Read/Write from Data Tables
Use these commands to access the bytes in Vision data tables without reference to table structure.

To determine the byte number of a data table cell, hold the cursor over the data table cell. A Tooltip opens, displaying the byte number.

Note ♦ A MODBUS command cannot read/write more than 242 DT bytes at one time.
♦ In addition, 0 is not a legal length.

Read from Data Table
Below, a MODBUS master reads data tables in Slave ID 1. Bytes 24-43 are read from Slave 1 into bytes 140-159 in the master's data tables.
### MODBUS

#### Parameter
- **Slave ID**: Constant or MI
  - The ID of the slave device containing the coils to be read (data source).

- **Slave: DT Start of Vector**: Constant, MI, ML, or DW
  - The start of the vector of bytes to be read (data source).

- **Slave: DT Offset in Vector**: Constant, MI, ML, or DW
  - Offset from the Slave: DT Start of Vector

- **Master: DT Start of Vector**: Constant, MI, ML, or DW
  - This is the start of a vector of bytes that will contain the data read from the slave.

- **Master: DT Offset in Vector**: Constant, MI, ML, or DW
  - Offset from the Master: DT Start of Vector

- **Read: DT Vector Length**: Constant or MI
  - The vector length.
  - **Note**: A MODBUS command cannot read/write more than 242 DT bytes at one time. In addition, 0 is not a legal length.

- **Status Messages**: MI
  - Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.

- **Total Sessions**: DW
  - This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.

- **Acknowledgements**: DW
  - This is the number of times the slave device answers.

#### Write to Data Table

Below, a MODBUS master writes to data tables in Slave ID 1. Bytes 140-159 are written from the master into bytes 24-43 in the slave’s data tables.
### Parameter Types and Functions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave ID</td>
<td>Constant or MI</td>
<td>The ID of the slave device to which the data will be written (data target).</td>
</tr>
<tr>
<td>Slave: DT Start of Vector</td>
<td>Constant, MI, ML, or DW</td>
<td>The start of the vector of bytes to be written into (data target).</td>
</tr>
<tr>
<td>Slave: DT Offset in Vector</td>
<td>Constant, MI, ML, or DW</td>
<td>Offset from the Slave: DT Start of Vector</td>
</tr>
<tr>
<td>Master: DT Start of Vector</td>
<td>Constant, MI, ML, or DW</td>
<td>This is the start of a vector of bytes, in the master, that will contain the data to be written to the slave (data source)</td>
</tr>
<tr>
<td>Master: DT Offset in Vector</td>
<td>Constant, MI, ML, or DW</td>
<td>Offset from the Master: DT Start of Vector</td>
</tr>
<tr>
<td>Read: DT Vector Length</td>
<td>Constant or MI</td>
<td>The vector length. <strong>Note</strong>: A MODBUS command cannot read/write more than 242 DT bytes at one time. In addition, 0 is not a legal length.</td>
</tr>
<tr>
<td>Status Messages</td>
<td>MI</td>
<td>Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.</td>
</tr>
<tr>
<td>Total Sessions</td>
<td>DW</td>
<td>This is the number of times the master PLC will attempt to access the slave device. Note that this is a simple incremental counter. Initialize it by storing 0 into the selected DW.</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>DW</td>
<td>This is the number of times the slave device answers.</td>
</tr>
</tbody>
</table>

### Configuring a MODBUS Slave Device

To configure a MODBUS slave device, follow these steps:

1. **Set Slave ID**: Determine the Slave ID from which the data will be read.
2. **Set Slave: DT Start of Vector**: Determine the start byte of the source vector to be read in the slave's data tables.
3. **Set Slave: DT Offset in Vector**: Contains the value of the offset from the source vector within the data tables.
4. **Set Master: DT Start of Vector**: Determines the start byte of the target vector used to store the slave data.
5. **Set Master: DT Offset in Vector**: Contains the value of the offset from the target vector within the data tables.
6. **Set Read: DT Vector Length**: Determines the length of the source vector, the number of bytes to be read from the slave's data tables.
7. **Set Status Messages**: Shows a message number. To check status and diagnose errors, check the MODBUS Status Messages.
8. **Set Total Sessions**: This is the number of times the master PLC will attempt to access the slave device. Initialize it by storing 0 into the selected DW.
9. **Set Acknowledgements**: This is the number of times the slave device answers.
MODBUS

The Ladder section below shows what elements are necessary to enable a master device to read from a slave. Note that the MODBUS Scan_EX operation should **not** be performed during the initial program scan.

Note that you must use a condition (RLO) to activate the MODBUS Configuration and SCAN_EX.

**At Power-up:**
- The Com port is initialized.
- MODBUS is configured.

**After Power-up the MODBUS Scan_EX FB enables a master device to read and write to/from the slave.**

### Slave Addressing

**Slave Address Tables**

<table>
<thead>
<tr>
<th>Pointer Value From:</th>
<th>Operand type</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>MB 0-2999</td>
<td>#01 Read Coils</td>
<td>#15 Force Coils</td>
</tr>
<tr>
<td>3000</td>
<td>SB</td>
<td></td>
<td>#15 Force Coils</td>
</tr>
<tr>
<td>4000</td>
<td>I (read-only)</td>
<td></td>
<td>Read-only</td>
</tr>
<tr>
<td>5000</td>
<td>O</td>
<td></td>
<td>#15 Force Coils</td>
</tr>
<tr>
<td>6000</td>
<td>T(readonly)</td>
<td></td>
<td>Read-only</td>
</tr>
</tbody>
</table>

Com Init and MODBUS configuration should be performed during the Power-up program scan.
MODBUS

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
<th>Modbus Command</th>
<th>Read</th>
<th>Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>7000</td>
<td>C (read-only)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8000</td>
<td>MB 3000-4095</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note ♦ Note that in order to access MBs 3000-4095, you address as follows: to access MB 3012, request slave address 8012.

<table>
<thead>
<tr>
<th>Registers</th>
<th>MODBUS Command Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pointer Value</td>
<td>Operand type</td>
</tr>
<tr>
<td>From:</td>
<td></td>
</tr>
<tr>
<td>0000</td>
<td>MI</td>
</tr>
<tr>
<td>4000</td>
<td>SI</td>
</tr>
<tr>
<td>5100</td>
<td>ML</td>
</tr>
<tr>
<td>6100</td>
<td>SL</td>
</tr>
<tr>
<td>6300</td>
<td>MDW</td>
</tr>
<tr>
<td>6700</td>
<td>SDW</td>
</tr>
<tr>
<td>6900</td>
<td>Timer preset</td>
</tr>
<tr>
<td>7200</td>
<td>Timer current</td>
</tr>
<tr>
<td>7500</td>
<td>Counter</td>
</tr>
<tr>
<td>7700</td>
<td>MF 0</td>
</tr>
</tbody>
</table>

Examples

The examples below show that:
- MODBUS addressing systems start at 1.
- Vision addressing start at 0.

**Bit Operands**
**MODBUS**

**Read** a 10-bit vector of inputs in a slave Vision controller, starting at Input 20, via Read Coils (MODBUS COMMAND #1)

- Vision PLC as the MODBUS master
  In VisiLogic's Read Coils FB, set the Slave: Start of Vector parameter to 4020, and the Read: Vector Length parameter to 10. Within the slave Vision controller, VisiLogic will read I 20 - I 29.

- SCADA as the MODBUS master
  In the SCADA application, set the Slave: Start of Vector parameter to 34021(30001 + 4000 + 20), and the Read: Vector Length to 10, enabling the Master device to read I 20 - I 29 within the slave Vision controller.

**Write** a 3-bit vector of outputs in a slave Vision controller, starting at Output 8, via Force Coils (MODBUS COMMAND #15)

- Vision PLC as the MODBUS master
  In VisiLogic's Force Coils FB, set the Slave: Start of Vector parameter to 5008, and the Read: Vector Length parameter to 3. Within the slave Vision controller, VisiLogic will write to O 8 - O 10.

- SCADA as the MODBUS master
  In the SCADA application, set the Slave: Start of Vector parameter to 35009 (30001 + 5000 + 8) and the Read: Vector Length parameter to 3, enabling the Master device to write to O 8 - O 10 within the slave Vision controller.

---

### Registers

**Read** a 9-register long vector of **16-bit integers** in a slave Vision controller, starting at MI 32, via Read Holding Registers (MODBUS COMMAND #03)

- Vision PLC as the MODBUS master
  In VisiLogic's Read Holding Registers FB, set the Slave: Start of Vector parameter to 32, and the Read: Vector Length parameter to 9. Within the slave Vision controller, VisiLogic will read MI 32 - MI 41.

- SCADA as the MODBUS master
  In the SCADA application, set the Slave: Start of Vector parameter to 40033 (40001 + 0000 + 3), and the Read: Vector Length parameter to 9, enabling the Master device to read MI 32 - MI 41 within the slave Vision controller.

**Note**

- If, within the Slave: Start of Vector parameter, the selected register type is a **32-bit double register** (slave addresses 5100 and greater) the preset vector length must be **doubled** as well.
  If, for example in the VisiLogic Preset Holding Registers FB:
    - Slave: Start of Vector parameter is set to 6300, and
    - You wish to preset 4 registers, for a total of 16 bytes
    - You must set the Preset Vector length to 8.

  Note that this means that, in these cases, the Preset: Vector Length parameter will always be an even number.

**Read** a 9-register long vector of **32-bit integers** in a slave Vision controller, starting at SL 32, via Preset Holding Registers (MODBUS COMMAND #16)

- Vision PLC as the MODBUS master
  In VisiLogic's Preset Holding Registers FB, set the Slave: Preset Vector parameter to 6132, and the Read: Vector Length parameter to **18** (2x9, in order
to fit the 32-bit SL registers). Within the slave Vision controller, VisiLogic will read SL 32 - SL 41.

- **SCADA as the MODBUS master**
  In the SCADA application, set the Slave: Start of Vector parameter to 406133, and the Read: Vector Length parameter to 18, enabling the Master device to read SL 32 - SL 41 within the slave Vision controller.

---

**Write a 6-register long vector of 16-bit integers** in a slave Vision controller, starting at MI 32, via Preset Holding Registers (MODBUS COMMAND #16)

- **Vision PLC as the MODBUS master**
  In VisiLogic's Preset Holding Registers FB, set the Slave: Start of Vector parameter to 32, and the Preset: Vector Length parameter to 6. Within the slave Vision controller, VisiLogic will write to MI 32 - MI 37.

- **SCADA as the MODBUS master**
  In the SCADA application, set the Slave: Start of Vector parameter to 40033, and the Read: Vector Length parameter to 6, enabling the Master device to write to MI 32 - MI 37 within the slave Vision controller.

---

**MODBUS via GSM or Standard Modem**

![Diagram of MODBUS via GSM or Standard Modem](image)

To run MODBUS via landline or GSM modem, use the appropriate SB in your MODBUS scan conditions to detect when the modem has finished initializing.

**Note**

When MODBUS (Serial) is configured to a port linked to a modem, the MODBUS function checks SI 100 Maximum Time Delay between characters (units 2.5ms) MODBUS + Modem. If SI 100 = 1, a time interval
of up to 2.5 msecs is permitted between characters, if SI 100 contains 2, the permitted interval is 5 msecs (n x 2.5 = interval. Note that the application must update SI 100 before the MODBUS configuration is activated.

MODBUS Status Operands

All of the Status operands linked to MODBUS FBs should be assigned Power-up Values; bits should be reset, and registers initialized to 0.

<table>
<thead>
<tr>
<th>MODBUS: Configuration FB Status Operand</th>
</tr>
</thead>
<tbody>
<tr>
<td>All MODBUS operations run through a MODBUS configuration placed in the master device's program.</td>
</tr>
</tbody>
</table>

**Function in Progress**

<table>
<thead>
<tr>
<th>Shows status of master's MODBUS Configuration</th>
</tr>
</thead>
</table>

**MB**

**Turns ON when:**

- A master Vision initiates MODBUS communication.
- Remains ON during the MODBUS session.

**Turns OFF when**

- The MODBUS: Configuration is activated.
- An answer is received from a slave.
- The TimeOut defined in the Configuration is exceeded.
- Certain Status Messages are given.

**MODBUS Operation Status Operands**

When you place MODBUS operations in your application (Force, Read, Preset, and Loopback commands), you link the operands below. These show the status of MODBUS sessions.

**Status Messages**

<table>
<thead>
<tr>
<th>Shows status of master's data requests and the replies the master receives from the slaves</th>
</tr>
</thead>
</table>

**MI**

- Automatically initialized to 0 when MODBUS operation is activated.
- Updated at the end of each attempt to communicate via MODBUS.
- Indicates status of MODBUS communications, according to the table below. Note that the current value always shows the most recent status.
<table>
<thead>
<tr>
<th>#</th>
<th>Status Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Status OK</td>
</tr>
</tbody>
</table>
| 1 | Unknown Command Number  
   This is received from the slave device. |
| 2 | Illegal Data Address  
   - Master: an invalid address is found by the master before a data request is sent to a slave. This may result, for example, when an MI is used to provide vector length.  
   - Slave: The slave notifies the master that the data request command includes invalid addresses.  
   - Slave--ScanEX: When ScanEX receives an input parameter in the 32-bit range (for example, 5100(ML)), it automatically takes double-register values. If, for example, ScanEX receives a Read Register(6) request for 5100, it returns the values in 5100 and 5101. If, however, ScanEX receives Read Register(6) request for 5101, it returns Error #2-- since 5101 provides the 'high' bytes of the 32-bit register, it is not a legal address. |
| 3 | Slave to Master: Illegal Data Type Quantity  
   Number of operands requested by user exceeds the maximum  
   **Note** ♦ A MODBUS command cannot read more than 124 16-bit integers, 62 double registers, 62 float registers, or 1900 bit operands at one time. In addition, 0 is not a legal vector length. |
| 4 | Master--Time Out  
   The amount of time the master will attempt to establish a MODBUS session |
| 5 | No Communication  
   The MODBUS session cannot be established. |

**Note** ♦ Messages 4 & 5. **TimeOut** and **Number of Retries** are defined in the **Configuration**.  
A Retry is an attempt to establish a MODBUS session.  
If, for example, TimeOut is defined as 2 seconds, and number of Retries as 3:  
- the controller will try to establish the session once, and will continue to try for 2 seconds.  
- If the first attempt fails, the **Status Message value will be 4**, Master TimeOut.  
- The controller will try twice more, for a total of 3 retries over 6 seconds.  
- If all attempts fail, the **Status Message value will be 5**.  
- If any attempt succeeds, the Status Message will be 0.  

* 6 | Master-slave data incorrectly synchronized |
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>*7</td>
<td>Master-slave data incorrectly synchronized</td>
</tr>
</tbody>
</table>
| 8 | **Master to application: Illegal Data Type Quantity**<br>Number of operands requested by user exceeds the maximum permitted for that FB operation in the master.  
**Note** ♦ A MODBUS command cannot read more than 124 16-bit integers, 62 double registers, 62 float registers, or 1900 bit operands at one time.  
in addition, 0 is not a legal vector length. |
| 9 | **Slave ID =0**<br>An attempt does to communicate with Slave ID 0. |
| *11 | Master-slave data incorrectly synchronized |

* Messages 6, 7, and 11 mean that the master has found incompatible elements in the data sent between master and slave.
<table>
<thead>
<tr>
<th>SI</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Maximum Time Delay between characters (units 2.5ms) MODBUS + Modem</td>
<td>When MODBUS (Serial) is configured to a port linked to a modem, the MODBUS function checks SI 100. If SI 100 = 1, a time interval of up to 2.5 msecs is permitted between characters. If SI 100 contains 2, the permitted interval is 5 msecs (n x 2.5 = interval.  Note that:  - The power-up value is 1,  - the application must update SI 100 before the MODBUS configuration is activated.</td>
</tr>
<tr>
<td>Index</td>
<td>C</td>
<td>M</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>communication</td>
<td>9, 18, 26</td>
<td>MODBUS .................................. 9, 18, 22, 26</td>
</tr>
<tr>
<td>Data Tables</td>
<td>...........................................18</td>
<td>network ....................................9, 18, 22, 26</td>
</tr>
</tbody>
</table>