Disclaimer

This document contains Honeywell proprietary information. Information contained herein is to be used solely for the purpose submitted, and no part of this document or its contents shall be reproduced, published, or disclosed to a third party without the express permission of Honeywell International Sàrl.

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any direct, special, or consequential damages. The information and specifications in this document are subject to change without notice.

Copyright 2014 - Honeywell International Sàrl
## Contents

### Getting started with Modbus controllers
- Devices supported by the Modbus interface ................................................................. 6
- Other documentation on Modbus .................................................................................. 7

### Modbus setup
- Architectures for Modbus ............................................................................................. 10
- Serial connection ............................................................................................................. 10
- Modbus Plus connection ............................................................................................... 12
- Modbus TCP connection ............................................................................................... 15
- Attaching a redundant Modbus device to an FTE network ........................................... 16
- Modbus function codes .................................................................................................. 16
- Modbus data tables ......................................................................................................... 17

### Setting up a Modbus controller
- Setting up a Modbus Plus network ............................................................................... 19
- Serial communication settings ...................................................................................... 19

### Modbus configuration and addressing
- Defining a Modbus channel .......................................................................................... 22
- Modbus channel Main properties .................................................................................. 22
- Modbus channel Port properties ................................................................................... 23
- Modbus channel Redundant Port properties ................................................................. 26
- Alternating behavior of redundant channels ................................................................. 26

### Modbus configuration and addressing
- Defining a Modbus controller ....................................................................................... 29
- Modbus controller Main properties .............................................................................. 29
- Manually defining a Modbus address for a point parameter ........................................... 33
- Entering an address for a Modbus controller .................................................................. 33
- Using Address Builder to enter a Modbus controller address ....................................... 33
- Address syntax for Modbus controllers ......................................................................... 33
- Optimizing scanning performance .................................................................................. 37
- Reducing the number of scan packets .......................................................................... 37
- Reducing the number of data tables being scanned ....................................................... 37

### Server tasks for Modbus
- Testing Modbus communications with the server ....................................................... 40
- Troubleshooting point configuration errors .................................................................... 41

### Notices
- Documentation feedback ............................................................................................... 44
- How to report a security vulnerability ............................................................................ 45
Getting started with Modbus controllers

This reference describes how to set up, configure, and test Modbus controllers.

**How to use this guide**

Complete each step before commencing the next step.

<table>
<thead>
<tr>
<th>Step</th>
<th>Go to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set up the controller and network</td>
<td>“Setting up a Modbus controller” on page 19</td>
</tr>
<tr>
<td>Use Quick Builder to define channels</td>
<td>“Defining a Modbus channel” on page 22</td>
</tr>
<tr>
<td>Use Quick Builder to define controllers</td>
<td>“Defining a Modbus controller” on page 29</td>
</tr>
<tr>
<td>Download channel and controller definitions to the server</td>
<td>For information about downloading to the server, see the topic titled &quot;Downloading items&quot; in the Quick Builder Guide.</td>
</tr>
<tr>
<td>Test communications</td>
<td>“Testing Modbus communications with the server” on page 40</td>
</tr>
<tr>
<td>Use Quick Builder to define points</td>
<td>“Manually defining a Modbus address for a point parameter” on page 33</td>
</tr>
</tbody>
</table>

**Related topics**

“Devices supported by the Modbus interface” on page 6
“Other documentation on Modbus” on page 7
Devices supported by the Modbus interface

The server supports the 584 and 984 Modbus controllers. The server also supports other controllers that use the Modbus protocol. However, there may be differences in the interpretation of addresses. For example, in some controllers, addresses begin at 0 instead of 1.

The server communicates with Modbus controllers by way of the Modbus RTU, Modbus ASCII, Modbus Plus, or Modbus TCP protocol.
Other documentation on Modbus

- Modbus Plus Network Planning and Installation Guide
Modbus setup

This section describes how to set up a Modbus controller.

Related topics
“Architectures for Modbus” on page 10
“Setting up a Modbus controller” on page 19
Architectures for Modbus

The server supports both single and redundant communications to Modbus controllers via serial connection, a Modbus Plus connection (Modbus Plus is Modbus's Local Area Network), or a Modbus TCP connection.

Serial connection

The server supports RS-232, RS-422, and RS-485 connection to Modbus controllers. Use RS-422 or RS-485 when you need to multi-drop more than one Modbus controller on a single channel.

Direct-server connection

![Diagram of RS-232 connection](image)

Figure 1: RS-232 connection

The following figures show the RS-232 wiring details for the two types of server connector.

![Diagram of RS-232 wiring details](image)

Figure 2: RS-232 wiring details for a server with a DB 9 connector
RS-232 wiring details for a server with a DB 25 connector

RS-422 connection

You use a Stallion EasyConnection serial adapter to multi-drop Modbus controllers via RS-422. Each controller must have a unique ID on the channel.

RS-485 connection

You use a Stallion EasyConnection serial adapter to multi-drop Modbus controllers via RS-485. Each controller must have a unique ID on the channel.
Modbus Plus connection

To connect Modbus controllers to the server through a Modbus Plus network, you can choose to use the following hardware:

- PCI-85 interface adapter, which is a PCI card installed on the server (Schneider Electric part number 416NHM30030A for single cable links, 416NHM30032A for dual cable links).
  
  You will also need the Modbus Plus drivers supplied with the PCI adapter.

- USB to Modbus Plus adapter (Schneider Electric part number TSXCUSBMBP Rev 2).
  
  You will also need the TSXCUSBMBP driver and Virtual Serial Port software supplied with USB adapter.

**Attention**

The PCI-85 interface adapters only work within servers that have PCI-X slots and will not work in PCI-E slots.

**Non-redundant Modbus Plus network architecture using a PCI-85 adapter**

If you require non-redundant communications, you use a single PCI-85 interface adapter:
If you require redundant communications, there are two options:

- Using a PCI-85 dual cable interface adapter (part number 416NHM30032A), which provides Modbus Plus cable redundancy. This is the most common redundant communications architecture.
- Using two PCI-85 single cable interface adapters (part number 416NHM30030A).
Non-redundant Modbus Plus network architecture using a USB adapter

If you require non-redundant communications, you use a single USB to Modbus Plus interface adapter:

Redundant Modbus Plus network architecture using USB adapters

If you require non-redundant communications, you need two USB to Modbus Plus interface adapters:
**Modbus Plus network routing**

The server supports the five-byte Modbus Plus network routing path. The five bytes allow controllers to be addressed that are up to four networks away. In the following figure, the ID for the Controller is **12.7**.

**Figure 13: An example Modbus Plus routed connection**

For more details about routing paths see the *Modbus Plus Network Planning & Installation Guide*.

---

**Modbus TCP connection**

To connect Modbus controllers to the server communicating using the Modbus TCP protocol, you are required to have network adapter (NA card) connected to an Ethernet network on both the Server as well as the Controller. In the case of a Modbus Quantum PLC, the NA card is referred to as an NOE module.
Redundant communication architecture

If you require redundant communications, you must have two separate network adapters (NA card) on both the Server and the Controller which are connected to separate Ethernet subnets. In the case of a Modbus Quantum PLC, the NA card is referred to as an NOE module.

Attaching a redundant Modbus device to an FTE network

Modbus Serial over FTE

Modbus controllers supporting serial connections can redundantly connect to an FTE network by using either a single terminal server or two separate terminal servers. It does not matter to which switch you connect the terminal servers, although to maximize redundancy, if you're using two terminal servers, you should connect one to the yellow switch and one to the green switch.

Connect the terminal server(s) to the primary and backup serial links on your Modbus device.

Modbus TCP over FTE

If your Modbus TCP devices supports redundant links in the same subnet, you should be able to connect the links directly to each switch (for example, primary link to the yellow switch and backup link to the green switch). Note that most Modbus TCP devices do not appear to support this topology.

If your Modbus TCP device does not support redundant links in the same subnet, you will need to set up a second subnet from your yellow NICs on your servers. To do so, in the Advanced property page for the TCP/IP protocol of the yellow NIC, add a second IP address using a private subnet. You can then connect the primary link on the Modbus device to one of the switches, and the backup link to the other.

Modbus function codes

The server supports the following Modbus function codes.
<table>
<thead>
<tr>
<th>Modbus Function Code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Read Coil/Status</td>
</tr>
<tr>
<td>02</td>
<td>Read Input Status</td>
</tr>
<tr>
<td>03</td>
<td>Read Holding Registers</td>
</tr>
<tr>
<td>04</td>
<td>Read Input Registers</td>
</tr>
<tr>
<td>05</td>
<td>Force Single Coil</td>
</tr>
<tr>
<td>06</td>
<td>Preset Single Register</td>
</tr>
<tr>
<td>16</td>
<td>Preset Multiple Registers</td>
</tr>
</tbody>
</table>

**Attention**

Normally, the Modbus interface uses function code 6 (Preset Single Register) to write data to registers within Modbus devices. However, some devices do not support function code 6, and so function code 16 (Preset Multiple Registers) must be used instead.

The **Use function code 16 to write Modbus device** check box on the controller’s Main tab in Quick Builder enables this feature. For more information about using function code 16, see the topic titled “Modbus controller Main properties.”

**Related topics**

“Modbus controller Main properties” on page 29

**Modbus data tables**

Modbus controllers store point parameter values in four data tables. The following table shows the read/write capabilities of these tables.

**Attention**

You need to create a “logical” controller in Quick Builder for each data table in a physical controller that you want to access. For example, if you want to read from a controller's Digital Input table and read from/write to its Input Register, you need to create two logical controllers. (Unless you have combined data within tables—see the “Reducing the number of data tables being scanned” topic.)

<table>
<thead>
<tr>
<th>Point Parameter</th>
<th>Data Table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Digital Output (read/write)</td>
</tr>
<tr>
<td>Analog</td>
<td>Yes</td>
</tr>
<tr>
<td>PV SOURCE</td>
<td>Yes</td>
</tr>
<tr>
<td>OP SOURCE</td>
<td>Yes</td>
</tr>
<tr>
<td>OP DESTIN</td>
<td>Yes</td>
</tr>
<tr>
<td>MD SOURCE</td>
<td>Yes</td>
</tr>
<tr>
<td>MD DESTIN</td>
<td>Yes</td>
</tr>
<tr>
<td>Ax SOURCE</td>
<td>Yes</td>
</tr>
<tr>
<td>Ax DESTIN</td>
<td>Yes</td>
</tr>
<tr>
<td>Status</td>
<td>PV SOURCE</td>
</tr>
<tr>
<td></td>
<td>OP SOURCE</td>
</tr>
</tbody>
</table>
### MODBUS SETUP

#### Data Table for Point Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Digital Output (read/write)</th>
<th>Digital Input (read only)</th>
<th>Input Register (read only)</th>
<th>Holding Register (read/write)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP DESTIN</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>MD SOURCE</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MD DESTIN</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td>Ax DESTIN</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accumulator</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>PV SOURCE</td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Related topics**

- “Reducing the number of data tables being scanned” on page 37
- “Defining a Modbus controller” on page 29
Setting up a Modbus controller

Related topics
“Setting up a Modbus Plus network” on page 19
“Serial communication settings” on page 19
“Modbus controller Main properties” on page 29

Setting up a Modbus Plus network

Installing the Modbus Plus adapter
Follow the installation instructions supplied with the Modbus Plus adapters.

⚠️ Attention
Remember to assign each Modbus Plus adapter with a unique Modbus+ address.

Installing the Modbus software
The latest version of the MBX drivers (MBX Driver Suite) can be downloaded from the Cyberlogic web site (http://www.cyberlogic.com). Follow the installation instructions supplied with the MBX Driver Suite.

Included with the MBX driver is a test utility, mbxdemo, which allows you to check that the adapter and device driver have been installed and configured correctly.

Serial communication settings

The DIP switch settings on the controller determine its communications characteristics. These settings are:

- Baud (9600)
- Parity (usually even)
- Checksum
- Protocol (RS-232)
- PLC Station ID (must be unique on the channel)

Record these settings—you need them when defining the controller and channel in Quick Builder.
Modbus configuration and addressing

This section describes how to configure a Modbus controller using Quick Builder.

Related topics
“Defining a Modbus channel” on page 22
“Defining a Modbus controller” on page 29
“Manually defining a Modbus address for a point parameter” on page 33
“Optimizing scanning performance” on page 37
Defining a Modbus channel

To define a Modbus channel using Quick Builder

1. Click \(\text{Add Item}\) to display the Add Items dialog.
2. Select Channel as the item and Modbus as the type.
3. Set the property values on the Main tab. See the topic titled "Modbus channel Main properties" for more information.
4. Click the Port tab and for Port Type select either:
   - \text{Serial}
   - \text{Terminal Server}
   - \text{LAN Vendor}

See the topic titled "Modbus channel Port properties" for more information.

To define a redundant Modbus channel using Quick Builder

1. Click the Redundant Port tab.
2. Select \text{Redundant Serial} or \text{LAN Vendor} as the Port Type.
3. Set the other properties as appropriate. See the topic titled "Modbus channel Redundant Port properties" for more information.

   The Redundant Port settings must be the same as the Port settings, except for the port name.

Related topics

- "Modbus channel Main properties" on page 22
- "Modbus channel Port properties" on page 23
- "Modbus channel Redundant Port properties" on page 26
- "Alternating behavior of redundant channels" on page 26

Modbus channel Main properties

The Main tab defines the basic properties for a Modbus channel.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces.</td>
</tr>
<tr>
<td>Description</td>
<td>(Optional) A description of the channel. A maximum of 30 alphanumeric characters, including spaces.</td>
</tr>
<tr>
<td>Marginal Alarm Limit</td>
<td>The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. To calculate an acceptable marginal alarm limit, use the formula: Square root of the number of controllers on the channel × Marginal Alarm Limit defined on those controllers (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to 25, the value would be (\sqrt{9} \times 25 = 75).</td>
</tr>
</tbody>
</table>
### Property | Description
--- | ---
Fail Alarm Limit | The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. Set this to double the value specified for the channel Marginal Alarm Limit.

Diagnostic Scan Rate | The period, in seconds, between diagnostic scans that verify communications integrity with the controller. The default value is 60 seconds. The diagnostic scans continue even if a controller is marked as failed, thus enabling the system to detect return-to-normal communications.

Write Delay | (Only applicable to Serial Port) The number of milliseconds that the server waits before writing to the controller.

Connect Timeout | The time, in seconds, the server attempts to connect to a controller before giving up. The default value is 10 seconds. For Modbus TCP protocol, it is recommended to use a timeout of 1 or 2 seconds, unless a loaded Ethernet network necessitates a higher timeout value.

Read Timeout | The time, in seconds, the server attempts to read data from a controller before giving up. The default value is 2 seconds. For Modbus TCP protocol, it is recommended to use a timeout of 1 second, unless a loaded Ethernet network necessitates a higher timeout value.

Item Type | Shows the type of item specified when this item was created.

Last Modified | Shows the date of the most recent modification to this channel's property details.

Last Downloaded | This shows the date that the item was last downloaded to the server.

Item Number | The unique item number currently assigned to this item in the format CHNnn. You can change the item number if you need to match your current server database configuration. The number must be between 1 and the maximum number of channels allowed for your system.

### Modbus channel Port properties

The **Port** tab defines the communication-related properties for a channel. The properties vary according to the selected **Port Type**:

- **Serial**. Select this if you are using a Stallion EasyConnection adapter or a RS-232 serial port. See the section below titled "Serial port properties."
- **TerminalServer**. Select this if you want to connect the controller to the server via a LAN. See the section below titled "TerminalServer port properties."
- **LANVendor**. Select this if you want to connect the controller to a Modbus Plus or Modbus TCP network. See the section below titled "LANVendor port properties."

**Attention**

- Set the port properties to the same values as those specified when configuring the controller.

#### Serial port properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial Port Name</td>
<td>The device name of the serial port.</td>
</tr>
</tbody>
</table>
### Modbus Configuration and Addressing

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baud</strong></td>
<td>The number of data bits per second. The default is 9600. This setting may be different if you are not using a 584 or 984 Modbus Controller.</td>
</tr>
<tr>
<td><strong>Number of Data Bits</strong></td>
<td>Set this to 8. This setting may be different if you are not using a 584 or 984 Modbus Controller.</td>
</tr>
<tr>
<td><strong>Stop Bits</strong></td>
<td>Set this to 1. This setting may be different if you are not using a 584 or 984 Modbus Controller.</td>
</tr>
<tr>
<td><strong>Parity</strong></td>
<td>Set this to <strong>EVEN</strong>. This setting may be different if you are not using a 584 or 984 Modbus Controller.</td>
</tr>
<tr>
<td><strong>Checksum</strong></td>
<td>Set this to <strong>None</strong>.</td>
</tr>
<tr>
<td><strong>XON/XOFF</strong></td>
<td>Set this to <strong>None</strong>.</td>
</tr>
<tr>
<td><strong>RS-232</strong></td>
<td>Only applicable if the controller is connected to the server via an RS-232 link.</td>
</tr>
<tr>
<td></td>
<td><strong>Enable RTS/CTS flow control.</strong> Select this if you want to use RTS/CTS for flow control to stop a receiver from being overrun with messages from a sender.</td>
</tr>
<tr>
<td></td>
<td><strong>Detect DCD.</strong> Select this if the Data Carrier Detect communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status line is not high—for example, on a dial-up link connection for a modem.</td>
</tr>
<tr>
<td></td>
<td><strong>Detect DSR.</strong> Select this if the Data Set Ready communication status line of the COM port requires monitoring (usually when using modem or microwave linking). When selected, the communications fails if the desired COM status is not achieved.</td>
</tr>
<tr>
<td><strong>RS-485</strong></td>
<td>Only applicable if the controllers are connected to the server via an RS-485 link.</td>
</tr>
<tr>
<td></td>
<td>Select <strong>Enable Stallion RS-485 Half Duplex</strong> and <strong>Echo</strong>. (Echo indicates that the server expects messages it sends to the port on the transmit line to be echoed back on the receive line.)</td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td>Modbus protocol to use when communicating with devices on this channel. The default is Modbus RTU. Select one of the following:</td>
</tr>
<tr>
<td></td>
<td>• Modbus RTU</td>
</tr>
<tr>
<td></td>
<td>• Modbus ASCII</td>
</tr>
</tbody>
</table>

### Terminal Server Port Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal Server TCP Host Name</td>
<td>The name and port number of the terminal server to which the channel is connected. You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Idle Timeout</td>
<td>The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of 0 indicates that the connection is never closed.</td>
</tr>
<tr>
<td>Checksum</td>
<td>Set this to none.</td>
</tr>
<tr>
<td>Protocol</td>
<td>Modbus protocol to use when communicating with devices on this channel. The default is Modbus RTU. Select one of the following: • Modbus RTU • Modbus ASCII</td>
</tr>
</tbody>
</table>

**LAN Vendor port properties**

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter Number</td>
<td>(Applicable to Modbus Plus Protocol only) Specify one of the following. See the section below titled “Specifying adapter number properties” for more information. • Board0 • Board1 • Board2 • or leave blank</td>
</tr>
<tr>
<td>Network Type</td>
<td>Modbus protocol to use when communicating with devices on this channel. The default is Modbus Plus. Select one of the following: • Modbus Plus • Modbus TCP If you select Modbus TCP, configure the IP address on the Controller Main tab.</td>
</tr>
</tbody>
</table>

**Specifying adapter number properties**

<table>
<thead>
<tr>
<th>Adapter Number Field Description</th>
<th>Corresponds To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Board0</td>
<td>Adapter 0 (Device 0)</td>
</tr>
<tr>
<td>Board1</td>
<td>Adapter 1 (Device 1)</td>
</tr>
<tr>
<td>Board2</td>
<td>Adapter 2 (Device 2)</td>
</tr>
</tbody>
</table>

If the Adapter Number is left blank:
• for a non-redundant channel, Adapter 0 is assumed
• for a redundant channel, Adapters 0 and 1 are assumed

If an Adapter number is specified, then a redundant channel cannot be configured. See the topic titled “Modbus channel Redundant Port properties” for more information on configuring redundant channels.

**Related topics**
“Defining a Modbus channel” on page 22
“Modbus channel Redundant Port properties” on page 26
Modbus channel Redundant Port properties

The Redundant Port tab defines the communication-related properties for a redundant channel. The redundant port settings, except for the port name, must be the same as for the Port tab. See the topic titled “Serial port properties” for more information.

**Attention**

For Modbus Plus, if you are using a PCI-85 (416NHM30032A) interface adapter that provides dual cable redundancy, you should not configure the channel as redundant, since redundancy is handled automatically by the interface adapter. You should only configure a Modbus Plus channel as redundant if you are using two single cable PCI-85 (416NHM30030A) network adapters.

---

**Related topics**

“Defining a Modbus channel” on page 22
“Modbus channel Port properties” on page 23

Alternating behavior of redundant channels

Alternating behavior of redundant channels is applicable only when the channel protocol for redundant channel is set to "Modbus TCP." Note that a channel or controller has to fail in order for a non-alternating channel to use the other link.

The alternating behavior of redundant Modbus channels is defined in the controller in Hardware Builder using the ALTERNATE keyword during download. For more information about using Hardware Builder, see the Hardware and Point Build Reference.

**If the Alternate keyword is Yes**

If ALTERNATE=YES (the default), requests will alternate between link A and B. Whenever a control is performed to a parameter (for example, changing the Set Point in Station), a write request will be performed to one link, followed by a read request for the same address on the alternate link. If the user is not performing any controls, then the read requests will be made at the requested update period and will alternate between links.

For example:

User changes Set Point in Station.
Write the value on link A.
Read the same address for confirmation on link B.
User changes Set Point in Station.
Write the value on link A.
Read the same address for confirmation on link B.
…
Server does a periodic scan:
Read the address on link A.
Server does a periodic scan:
Read the address on link B.
Server does a periodic scan:
Read the address on link A.
…
User changes Set Point in Station.
Write the value on link B.
Read the same address for confirmation on link A.

If the Alternate keyword is No
If ALTERNATE=NO, then all requests will be made on the same link until the link fails.
For example:

User changes Set Point in Station.
Write the value on link A.
Read the same address for confirmation on link A.
User changes Set Point in Station.
Write the value on link A.
Read the same address for confirmation on link A.
…
Server does a periodic scan:
Read the address on link A.
Server does a periodic scan:
Read the address on link A.
Server does a periodic scan:
Read the address on link A.
…
Link A is marked as failed.
…
User changes Set Point in Station.
Write the value on link B.
Read the same address for confirmation on link B.
User changes Set Point in Station.
Write the value on link B.
Read the same address for confirmation on link B.
…
Server does a periodic scan:
Read the address on link B.
Server does a periodic scan:
Read the address on link B.
Server does a periodic scan:
Read the address on link B.
Related topics

“Defining a Modbus channel” on page 22
Defining a Modbus controller

In the current context, a “controller” is a logical controller that corresponds to a data table within a physical controller.

You need to define a “logical” controller for each Modbus data table (see the topic titled “Modbus data tables”) in the physical controller that you need to access. (Unless you have combined data within tables. See the topic titled “Reducing the number of data tables being scanned” for more information.)

To define a Modbus controller using Quick Builder

1. Click to display the Add Items dialog.
2. Select Controller as the item and Modbus as the type.
3. Set the property values on the Main tab. See the topic titled "Modbus controller Main properties" for more information.

Related topics
“Reducing the number of data tables being scanned” on page 37
“Modbus data tables” on page 17

Modbus controller Main properties

The Main tab defines the basic properties for a Modbus controller.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_ _) appear as spaces. For LAN–connected controllers, the name is used to look up the IP address in the TCP/IP database if you do not specify an IP Address property. In the case of communications redundancy when the IP Addresses are not defined in Quick Builder, the IP Address 1 and 2 must be specified in the Server hosts file. The host name for IP Address 1 is then the Name property with an “A” appended to it and the host name for IP Address 2 is the Name property with a “B” appended to it.</td>
</tr>
<tr>
<td>Description</td>
<td>(Optional) A description of the controller. A maximum of 30 alphanumeric characters, including spaces.</td>
</tr>
<tr>
<td>Channel Name</td>
<td>The Modbus channel on which the controller communicates with the server.</td>
</tr>
<tr>
<td>Protocol</td>
<td>The protocol of the channel used by the controller.</td>
</tr>
<tr>
<td>Skip CR/LF</td>
<td>Shows whether the device skips the CR/LF characters at the end of each message.</td>
</tr>
<tr>
<td>PLC Station ID</td>
<td>ID of the controller, specified when configuring the controller—see the topic titled “Setting up a Modbus controller.”</td>
</tr>
</tbody>
</table>
### MODBUS CONFIGURATION AND ADDRESSING

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Table</td>
<td>The data table that this controller represents. The address ranges for the data tables are as follows:</td>
</tr>
<tr>
<td></td>
<td>• Digital Output (read/write). 000001 to 065535</td>
</tr>
<tr>
<td></td>
<td>• Digital Input (read only). 100001 to 165535</td>
</tr>
<tr>
<td></td>
<td>• Input Register (read only). 300001 to 365535</td>
</tr>
<tr>
<td></td>
<td>• Holding Register (read/write). 400001 to 465535</td>
</tr>
<tr>
<td></td>
<td>The addressable range for point parameter data is always 00001 to 32766 within each individual table. To access addresses in the range 32767 to 65535, you need to define a logical controller with an appropriate offset value.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When you enter an address, do not include the table number prefix.</td>
</tr>
<tr>
<td>Offset</td>
<td>The offset for the address range. Offset + 1 is the first valid register. The offset must meet the following conditions:</td>
</tr>
<tr>
<td></td>
<td>• Offset is less than the lowest register of the device that you want to read.</td>
</tr>
<tr>
<td></td>
<td>• Offset + 2 is a single-register address. The server diagnostic scan polls this register. Any error returned for the diagnostic scan increments the barometer level.</td>
</tr>
<tr>
<td></td>
<td>The default value is 0. Therefore, by default, the first valid register is 00001, and the default register for the diagnostic scan is 00002.</td>
</tr>
<tr>
<td></td>
<td>If you want to use an address range between 32767 and 65535, you must specify a negative offset. For more information on calculating a negative offset, see the section below titled “Offset examples.”</td>
</tr>
<tr>
<td></td>
<td>The diagnostic address determined by the Offset value is overridden if the Diagnostic Address is specified. See the section below titled “Diagnostic address examples” for more information.</td>
</tr>
<tr>
<td></td>
<td><strong>Attention</strong> Once you have set an offset and downloaded the controller definition to the server, you cannot change the offset value. To change the offset value, you need to delete the controller and build it again.</td>
</tr>
<tr>
<td>Diagnostic Address</td>
<td>(Optional) The address in the controller to read for the diagnostic scan. This address and the next location will be read. For example, if 100 is specified, 100 and 101 will be read.</td>
</tr>
<tr>
<td></td>
<td>If specified, overrides the diagnostic address determined by the Offset value. See the section below titled “Diagnostic address examples” for more information.</td>
</tr>
<tr>
<td></td>
<td>If this is not specified, the Offset value is read.</td>
</tr>
<tr>
<td>IP Address 1 and 2</td>
<td>(Applicable to Modbus TCP Protocol only)</td>
</tr>
<tr>
<td></td>
<td>Network address(es) of the controller to connect to. IP Address 2 is used only when the Controller has a redundant network adapter card and the Channel that the Controller has been built on has been defined as a redundant channel.</td>
</tr>
<tr>
<td></td>
<td>If the IP Address is not specified, the Controller name is used as the TCP host name. For further information, see the Name property.</td>
</tr>
<tr>
<td></td>
<td>You can specify the port number to use. The ability to define a specific port enables multiple Modbus devices to be addressed behind a single IP address. If no port number is specified, port number 502 is used by default.</td>
</tr>
<tr>
<td>Item Type</td>
<td>Shows the type of item specified when this item was created.</td>
</tr>
<tr>
<td>Last Modified</td>
<td>Shows the date of the most recent modification to this controller's property details.</td>
</tr>
<tr>
<td>Last Downloaded</td>
<td>This shows the date that the item was last downloaded to the server.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Item Number</td>
<td>The unique item number currently assigned to this item in the format RTUnnn. You can change the item number if you need to match your current server database configuration. The number must be between 1 and the maximum number of controllers allowed for your system.</td>
</tr>
</tbody>
</table>

**Offset examples**

This example shows the hardware registers (including the diagnostic scan register) used when you specify an Offset value and when you don't specify a Diagnostic Address value.

<table>
<thead>
<tr>
<th>Offset</th>
<th>First hardware register</th>
<th>Diagnostic scan register</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>50</td>
<td>51</td>
<td>52</td>
</tr>
<tr>
<td>-30000</td>
<td>35536</td>
<td>35537</td>
</tr>
</tbody>
</table>

If you want to use an address range between 32767 and 65535, you must specify a negative offset and this value is calculated using the formula:

\[
\text{offset} = \text{first hardware register} - 65536
\]

For a diagnostic scan register that is not the next consecutive hardware register address from the first hardware register, you also need to specify a Diagnostic Address value.

**Diagnostic address examples**

This example shows the diagnostic scan registers used when you specify an Offset value and a Diagnostic Address value.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Diagnostic Address</th>
<th>Diagnostic scan register</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 or not specified</td>
<td>2</td>
</tr>
<tr>
<td>50</td>
<td>0 or not specified</td>
<td>52</td>
</tr>
<tr>
<td>0</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>50</td>
<td>20</td>
<td>71</td>
</tr>
<tr>
<td>-30000</td>
<td>0 or not specified</td>
<td>35537</td>
</tr>
<tr>
<td>-30000</td>
<td>200</td>
<td>35737</td>
</tr>
</tbody>
</table>

For diagnostic scan registers in the range between 32767 and 65535, the diagnostic scan register is calculated using one of the following formulas:

\[
\text{diagnostic scan register} = \begin{cases} 
65535 + \text{offset} + \text{diagnostic address} & \text{if the diagnostic address is greater than zero} \\
65537 + \text{offset} & \text{if the diagnostic address is zero}
\end{cases}
\]

You can calculate the diagnostic address using the formula:

\[
\text{diagnostic address} = \text{diagnostic scan register} - 65535 + \text{offset}
\]

For example, you need to build a point that scans the hardware address 40001 and for diagnostic scans you need to read the registers with the hardware address of 50000-50001.

To allow a specified address of 1 to represent the hardware address 40001, you calculate the offset using the following equation: \( \text{offset} = \text{first hardware register to use} - 65536 \). Therefore, \( \text{offset} = 40001 - 65536 \), which is -25525.

To calculate the diagnostic address, use the formula: \( \text{diagnostic address} = \text{diagnostic scan register} - 65535 - \text{offset} \). Therefore, \( \text{diagnostic address} = 50000 - 65535 - (-25525) \), which is 10000.
Thus you would need to build a controller with an offset of -25535, a diagnostic address of 10000 and a point with a specified address of 1.

**Hardware registers and the address to use in Quick Builder**

<table>
<thead>
<tr>
<th>Attention</th>
</tr>
</thead>
<tbody>
<tr>
<td>The calculations within this section are for negative offset values only.</td>
</tr>
</tbody>
</table>

To calculate the address to use for a point in Quick Builder, use the formula:

\[
\text{address in Quick Builder} = \text{hardware register} + 65535 - \text{offset}
\]

For example, if the offset is -25536 and you want to address register 40127, the address you need to use for the point in Quick Builder is based on the formula: \(\text{address in Quick Builder} = \text{hardware register} + 65535 - \text{offset}\). Therefore, \(\text{address in Quick Builder} = 40127 - 65535 - (-25536)\), which is 128.

To calculate the hardware address based on the address in Quick Builder, use the formula:

\[
\text{hardware register} = 65535 + \text{offset} + \text{address in Quick Builder}
\]

**Related topics**

“Modbus function codes” on page 16
“Setting up a Modbus controller” on page 19
Manually defining a Modbus address for a point parameter

This section describes how to manually define a Modbus address for a point parameter.

Tip
If you need to define multiple addresses, it is recommended that you use the Modbus configuration wizard.

Related topics
“Entering an address for a Modbus controller” on page 33
“Using Address Builder to enter a Modbus controller address” on page 33
“Address syntax for Modbus controllers” on page 33

Entering an address for a Modbus controller

For PV Source Address, Source Address, and Destination Address the format for a Modbus controller address is:

\[\text{ControllerName} \, \text{Address}\]

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ControllerName</td>
<td>The name of the Modbus controller.</td>
</tr>
<tr>
<td>Address</td>
<td>The address within the controller where the value is stored.</td>
</tr>
<tr>
<td></td>
<td>See the topic titled “Address syntax” for more information.</td>
</tr>
</tbody>
</table>

For help when defining an address, click \(\text{next to Address}\) to display Address Builder.

Using Address Builder to enter a Modbus controller address

To specify a Modbus controller address, specify the items as follows:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Type</td>
<td>Select \textit{Controller}.</td>
</tr>
<tr>
<td>Controller</td>
<td>Select the appropriate Modbus controller.</td>
</tr>
<tr>
<td>Address</td>
<td>The address within the controller where the value is stored.</td>
</tr>
<tr>
<td></td>
<td>See the topic titled &quot;Address syntax&quot; for more information.</td>
</tr>
</tbody>
</table>

Address syntax for Modbus controllers

The format for the address is:

\[\text{Address} \, [\text{DataFormat} \mid \text{BitNumber}]\]

<table>
<thead>
<tr>
<th>Part</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>The address (decimal) for the parameter value within the controller table.</td>
</tr>
<tr>
<td></td>
<td>If the address is prefixed with the table number, omit the table number.</td>
</tr>
<tr>
<td></td>
<td>For example, the first register in table 4 is often written as 40001 or 400001.</td>
</tr>
<tr>
<td></td>
<td>Enter 2 as the address.</td>
</tr>
<tr>
<td></td>
<td>The table number prefix is shown on the Scanning tab of the Point Detail display and when you use the commands listag, rtusum, and lisscn.</td>
</tr>
</tbody>
</table>
**Data format for scaling 16-bit integers**

You can scale point parameter values with a data format if they have addresses in the Input Register and Holding Register data tables. Select the format that corresponds to the counts that have been set in the controller register.

<table>
<thead>
<tr>
<th>Data Format</th>
<th>Counts in Controller Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>U4095 (default)</td>
<td>0–4095</td>
</tr>
<tr>
<td>U9999</td>
<td>0–9999</td>
</tr>
<tr>
<td>U999</td>
<td>0–999</td>
</tr>
<tr>
<td>U15B</td>
<td>0–32767</td>
</tr>
<tr>
<td>U16B</td>
<td>0–65535</td>
</tr>
<tr>
<td>S16B</td>
<td>-32768–32767</td>
</tr>
</tbody>
</table>

**Attention**

If auxiliary parameters have a data format type that requires scaling (U4095, U999, and so on), they take the same range as the PV.

**Data format for scaling 32-bit integers**

The Modbus Interface will combine two 16-bit registers (as defined by the Modbus protocol standard) into a 32-bit value.
The little-endian word formats listed in the following table.

<table>
<thead>
<tr>
<th>Data Format</th>
<th>Description</th>
<th>Counts in Controller Register</th>
</tr>
</thead>
<tbody>
<tr>
<td>S32B</td>
<td>Signed 32-bit Binary Little-endian</td>
<td>-2147483648–2147483647</td>
</tr>
<tr>
<td>U32B</td>
<td>Unsigned 32-bit Binary Little-endian</td>
<td>0–4294967295</td>
</tr>
<tr>
<td>S32BS</td>
<td>Signed 32-bit Binary Scaled Little-endian</td>
<td>-2147483648–2147483647</td>
</tr>
<tr>
<td>U32BS</td>
<td>Unsigned 32-bit Binary Scaled Little-endian</td>
<td>0–4294967295</td>
</tr>
</tbody>
</table>

These data formats assume the following word/byte order:
• two 16-bit registers in big endian byte order
• the first register transmitted is the low word
• the second register transmitted is the high word
The value is calculated as (high register x 65536) + low register.

The big-endian word formats listed in the following table.

<table>
<thead>
<tr>
<th>Data Format</th>
<th>Description</th>
<th>Counts in Controller register</th>
</tr>
</thead>
<tbody>
<tr>
<td>U32BB</td>
<td>Unsigned 32-bit Binary Big-endian</td>
<td>0–4294967295</td>
</tr>
<tr>
<td>U32BSB</td>
<td>Unsigned 32-bit Binary Scaled Big-endian</td>
<td>0–4294967295</td>
</tr>
<tr>
<td>S32BB</td>
<td>Signed 32-bit Binary Big-endian</td>
<td>-2147483648–2147483647</td>
</tr>
<tr>
<td>S32BSB</td>
<td>Signed 32-bit Binary Scaled Big-endian</td>
<td>-2147483648–2147483647</td>
</tr>
</tbody>
</table>

These data formats assume the following word/byte order:
• two 16-bit registers in big endian byte order
• the first register transmitted is the high word
• the second register transmitted is the low word
The value is calculated as (high register x 65536) + low register.

**Data format for reading floating point values**

Select format *IEEEFP* to read two consecutive registers as a single precision floating point number.

Variations on the IEEEFP format that you can specify include:

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEEFPB</td>
<td>Bytes are big endian format (this is the same as IEEEFP)</td>
</tr>
<tr>
<td>IEEEFPBB</td>
<td>Bytes are byte-swapped big endian format</td>
</tr>
<tr>
<td>IEEEFPL</td>
<td>Bytes are little endian format</td>
</tr>
<tr>
<td>IEEEFPLB</td>
<td>Bytes are byte-swapped little endian format</td>
</tr>
</tbody>
</table>

**Attention**

• IEEEFP numbers use two data addresses, *address* and *address-1*. Do not specify *address-1* (the lower address) as the point parameter location address.
• When configuring controller data tables, do not assign overlapping data addresses if floating point values are used and do not set an IEEEFP address to 1.
**Data format for reading raw values without scaling**

Select a format of C16, or enter a 0, to read all 16 bits in the register without scaling.

To read less than 16 bits without scaling enter the starting bit number (1 to 15).

If you are not using scaling, the point range is still used for PV indicator bar height only (the PV indicator bar is on the Point detail display on Station).
Optimizing scanning performance

A Modbus scan packet:

- Can contain a maximum of either 125 controller register addresses or 2,000 (125*16) controller digital addresses. (Two controller data tables have 16-bit registers: Input Register table and Holding Register table.)
- Must have addresses that reside within the same logical controller (that is, the same data table).
- Must have addresses with the same scan period.

Two types of scan packet are built for Modbus:

- **Hardware Diagnostic.** One scan per controller at a defined regular interval (the default is 60 seconds) to verify communications integrity with the controller. This scan packet is automatically created.

- **Periodic Data Acquisition.** A defined regular interval in which the server database acquires information from the data tables in the controller and processes the values as point parameters.

  You need to define the scan period for each point parameter source address. The scan period should reflect both the rate at which the value held in memory changes and its importance to the process (critical or non-critical).

  There is one periodic data acquisition scan per scan packet.

Reducing the number of scan packets

If the number of scan packets becomes too great, scanning performance is impaired.

To minimize the number of scan packets, use a small number of available scan periods for all your point definitions.

Closely block the registers read by the server and ensure that all addresses within a block (125 registers) have the same scan period.

Make each Modbus scan packet as close to the maximum size as possible and ensure there are no small packets being scanned at fast rates.

You verify your scanning strategy by using the List Scan utility, lisscn, to list the scan packets you have built.

Reducing the number of data tables being scanned

You can reduce the number of scan packets by combining data within data tables.

---

**Attention**

Combining data within tables requires complex controller logic and should be used only when other ways of improving scanning performance have failed.

---

You can write controller logic in such a way that pulls values from the Digital tables and the Register Input table into registers in the Holding Register table. In this way, you can reduce the number of logical controllers you need to define from four to one (the Holding register table only).

**Related topics**

“Modbus data tables” on page 17

“Defining a Modbus controller” on page 29
Server tasks for Modbus

This section describes tasks for the Modbus that you perform on the server.

Related topics

“Testing Modbus communications with the server” on page 40
“Troubleshooting point configuration errors” on page 41
Testing Modbus communications with the server

You use the diagnostic utility, **modtst**, to test communications between the Modbus controller and the server. Modbus controllers are fitted with transmit and receive diagnostic LEDs. If a communications fault illuminates an LED, see the Modbus manuals that came with your controllers.

**To run the modtst utility**

1. Open a Command Prompt window.
2. Type `modtst` and press ENTER.
   
   You can read and/or write data addressable by the server (see Modbus Data Tables and Address Range).

   To display help, type `?` and press ENTER.
3. When prompted, enter the relevant information.
Troubleshooting point configuration errors

If any errors are encountered as a result of running the **modtst** utility, review the steps for connecting and configuring a Modbus controller and ensure that definitions for channels and controllers match the settings in the controller itself.

**Diagnostic check**

*Error code 0106 (Device Timeout)*

**Cause**

No response was received from the controller.

**Solution**

Ensure that your channel and controller definitions match the controller's settings.
Notices

Trademarks
Experion® and SafeBrowse® are registered trademarks of Honeywell International, Inc.

Other trademarks
Microsoft and SQL Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Trademarks that appear in this document are used only to the benefit of the trademark owner, with no intention of trademark infringement.

Third-party licenses
This product may contain or be derived from materials, including software, of third parties. The third party materials may be subject to licenses, notices, restrictions and obligations imposed by the licensor. The licenses, notices, restrictions and obligations, if any, may be found in the materials accompanying the product, in the documents or files accompanying such third party materials, in a file named third_party_licenses on the media containing the product, or at http://www.honeywell.com/ps/thirdpartylicenses.
You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

http://www.honeywellprocess.com/support

If you have comments about Honeywell Process Solutions documentation, send your feedback to:

hpsdocs@honeywell.com

Use this email address to provide feedback, or to report errors and omissions in the documentation. For immediate help with a technical problem, contact your local Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.
How to report a security vulnerability

For the purpose of submission, a security vulnerability is defined as a software defect or weakness that can be exploited to reduce the operational or security capabilities of the software.

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services. To report a potential security vulnerability against any Honeywell product, please follow the instructions at: https://honeywell.com/pages/vulnerabilityreporting.aspx

Submit the requested information to Honeywell using one of the following methods:

• Send an email to security@honeywell.com.
  
or
• Contact your local Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.
Index

A
Address Builder
  Modbus controllers 33
  addressing
  Modbus controllers 33
architectures
  Modbus 10

C
channels
defining 22
  Modbus channel 22
commands and utilities
  modtst 40
communications
testing
  Modbus controllers 40
configuring
  Modbus controllers 5
connect timeout 22
connections
  Modbus Plus 12
  RS-422 10
  RS-485 10
  serial 10
controllers
  configuring
    Modbus controllers 5
  Modbus controllers 5, 29, 33

data table 29
defining
  controllers 29
  Modbus channel 22
  Modbus controllers 29
defining a Modbus address for a point parameter 33
devices supported 6
diagnostic address 29
diagnostic scan rate 22
diagnostics
  Modbus controllers 40
documentation 7

fail alarm limit 22, 29
function codes 16

H
hardware registers 29

M
Main tab
  Modbus channel 22
  Modbus controller 29
marginal alarm limit 22, 29
Modbus controllers
  address syntax 33
  addressing 33
  architecture 10
  channels 22
  configuring 5
  connect timeout 22
  controllers 29
  data table 29
  defining 29
  devices supported 6
  diagnostic address 29
  diagnostic scan rate 22
  fail alarm limit 22, 29
  function codes 16
  getting started 5
  hardware registers 29
  IP address 29
  marginal alarm limit 22, 29
  modtst utility 40
  offset 29
  protocol 29
  read timeout 22
  skip CR/LF 29
  supported devices 6
  testing communications 40
  troubleshooting 40, 41
  write delay 22
Modbus Plus
  connections 12
  network routing 12
Modbus TCP, connection 15
Modicon controllers
  , See Modbus
  modtst utility 40

offset 29

point parameters
defining an address 33
Port properties
  Modbus channel 23
  protocol 29

Redundant Port properties
  Modbus channel 26
  RS-422
    connections 10
  RS-485
    connections 10

scanning, optimizing performance 37
serial connections
  Modbus 10

skip CR/LF 29
supported devices 6

terminal server properties for a Modbus controller 23
testing communications
  Modbus controllers 40
troubleshooting
  Modbus controllers 41

utilities
  modtst 40

write delay 22