Honeywell

Experion PKS
Modbus Interface Reference

EPDOC-XX71-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-XX71-en-431A	431	0	February 2015

Disclaimer

Contents

Planning considerations for installing and configuring Modbus controllers

This reference describes how to set up, configure, and test Modbus controller communications with the server.

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

How to use this guide

The following steps show the order in which the controller interface should be configured. Complete each step before starting the next.

Steps for connecting and configuring a Modbus controller.

Step	Go to
Set up the controller and network	Setting up a Modbus Plus network
Use Quick Builder to define channels	• Modbus channel and controller reference • *Quick Builder User's Guide*
Use Quick Builder to define controllers	• Modbus channel and controller reference • *Quick Builder User's Guide*
Download channel and controller definitions to the server	*Quick Builder User's Guide*
Test communications	Testing Modbus communications with the server
Use Quick Builder to define points	Modbus points reference

Related topics

"Devices supported by the Modbus interface" on page 6
"Other documentation for Modbus" on page 7
"Architectures for Modbus" on page 8
"Setting up a Modbus Plus network" on page 17
"Serial communication settings for Modbus" on page 18
"Modbus channel and controller reference" on page 19
"Modbus points reference" on page 33
"Testing Modbus communications with the server" on page 38

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 for Modbus

- *Modbus Plus Network Planning and Installation Guide*

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 connections for Modbus

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

Direct-server connection

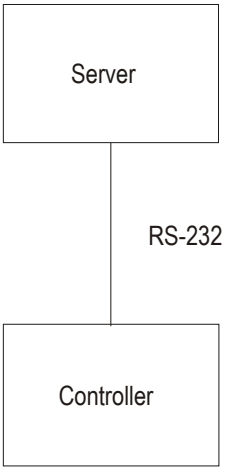

Figure 1: RS-232 connection

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

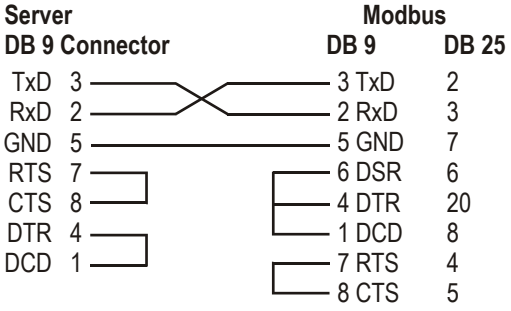

Figure 2: RS-232 wiring details for a server with a DB 9 connector

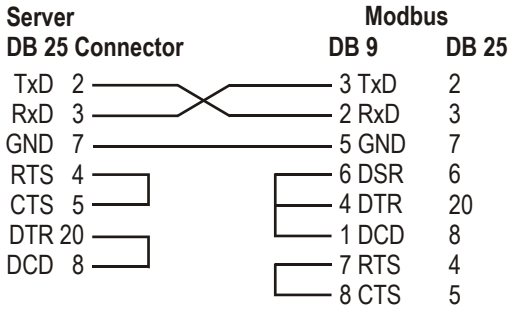

Figure 3: RS-232 wiring details for a server with a DB 25 connector

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.

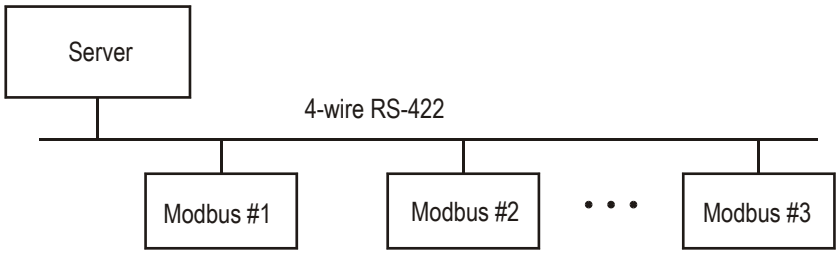

Figure 4: RS-422 multi-drop connection

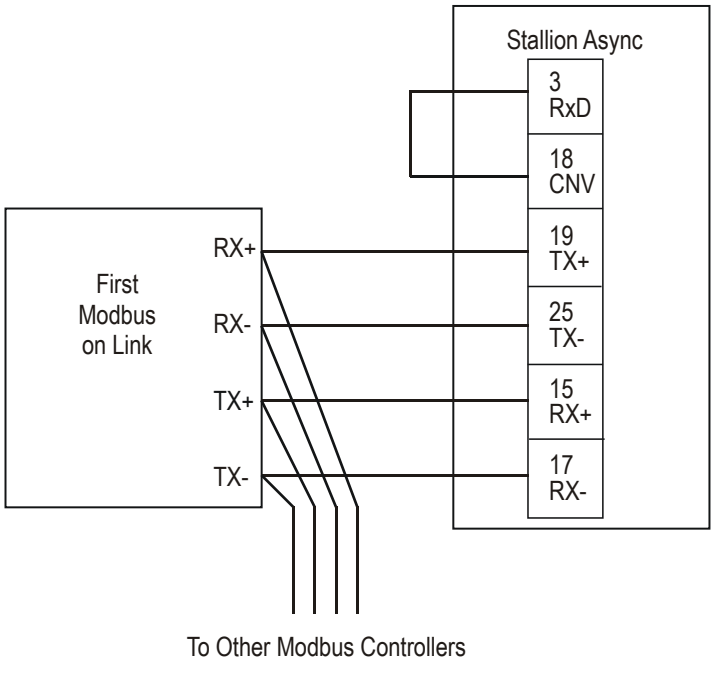

Figure 5: RS-422 wiring details

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.

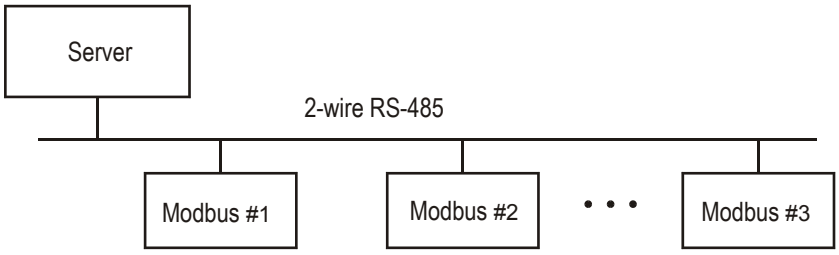

Figure 6: RS-485 multi-drop connection

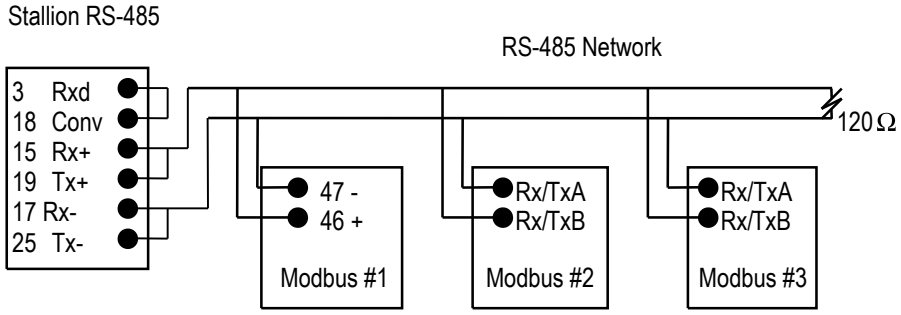

Figure 7: RS-485 wiring details

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:

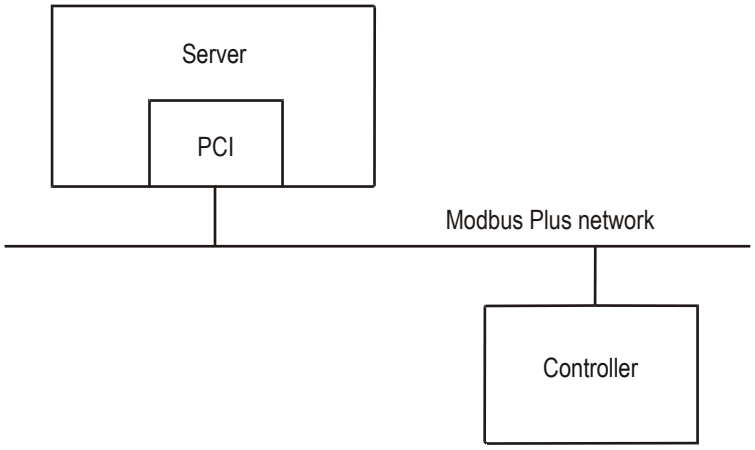

Figure 8: Non-redundant Modbus Plus network architecture using a PCI-85 adapter

Redundant Modbus Plus network architecture using PCI-85 adapters

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).

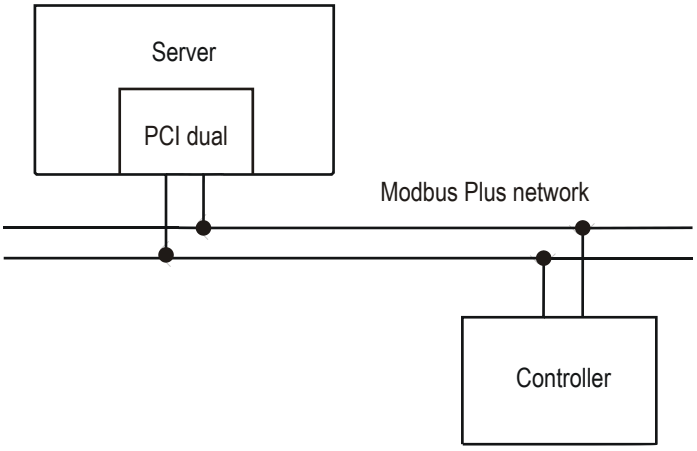

Figure 9: Redundant Modbus Plus network architecture using a PCI-85 adapter (416NHM30032A)

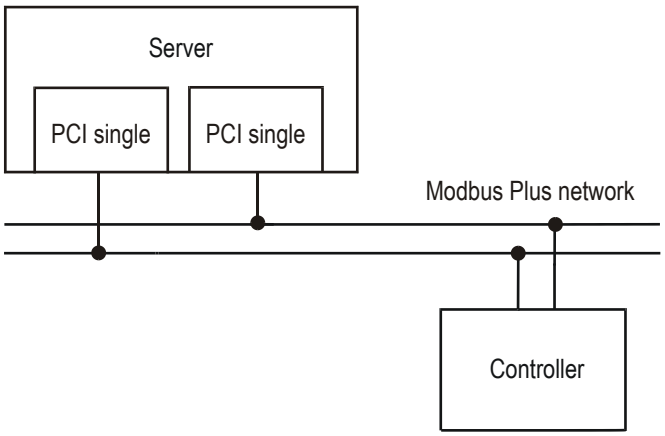

Figure 10: Redundant Modbus Plus network architecture using two PCI-85 adapters (416NHM30030A)

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:

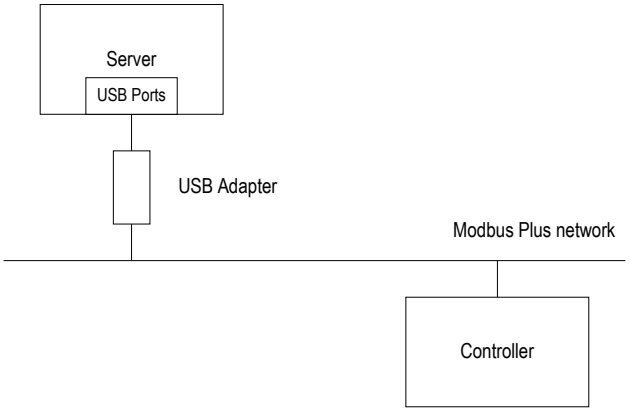

Figure 11: Non-redundant Modbus Plus network architecture using a USB to Modbus Plus adapter

Redundant Modbus Plus network architecture using USB adapters

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

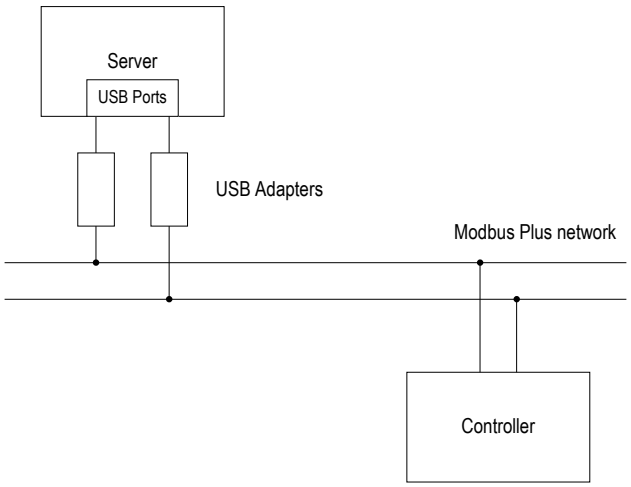

Figure 12: Redundant Modbus Plus network architecture using two USB to Modbus Plus adapters

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*.

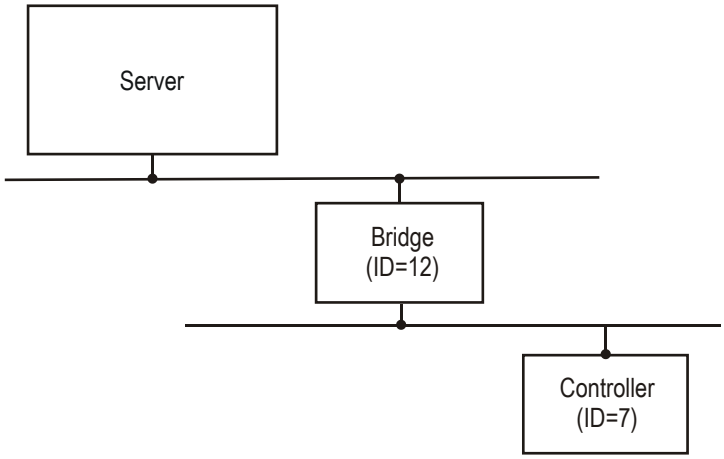

Figure 13: An example Modbus Plus routed connection

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 a *network interface card* (NIC) connected to an Ethernet network on both the Server as well as the Controller. In the case of a Modbus Quantum PLC, the NIC is referred to as a NOE module.

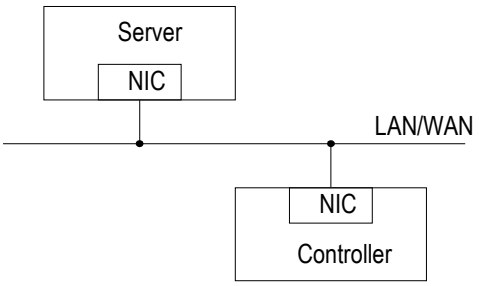

Figure 14: Non-redundant Modbus TCP network architecture

Redundant communication architecture

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

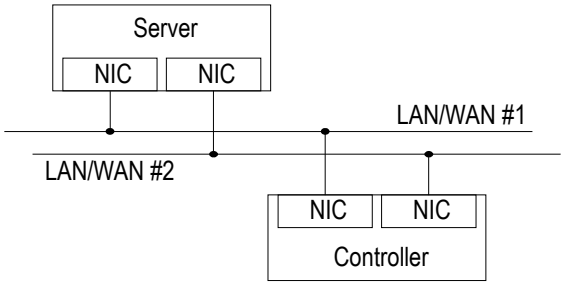

Figure 15: Redundant Modbus TCP network architecture

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.

Function codes supported by Modbus

The server supports the following Modbus function codes.

Modbus Function Code	Meaning
01	Read Coil/Status
02	Read Input Status
03	Read Holding Registers
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register
16	Preset Multiple Registers

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 "Main properties for a Modbus controller."

Related topics

"Main properties for a Modbus controller" on page 28

Data tables supported by Modbus

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 topic titled "Reducing the number of data tables being scanned for Modbus controllers" for more information.)

Point Parameter	Data Table			
	Digital Output (read/ write)	Digital Input (read only)	Input Register (read only)	Holding Register (read/ write)
Analog				
PV SOURCE			Yes	Yes
SP SOURCE			Yes	Yes
SP DESTIN				Yes
OP SOURCE			Yes	Yes
OP DESTIN				Yes
MD SOURCE	Yes	Yes	Yes	Yes
MD DESTIN	Yes			Yes
Ax SOURCE			Yes	Yes
Ax DESTIN				Yes
Status				
PV SOURCE	Yes	Yes	Yes	Yes
OP SOURCE	Yes	Yes	Yes	Yes

Point Parameter	Data Table			
	Digital Output (read/ write)	Digital Input (read only)	Input Register (read only)	Holding Register (read/ write)
OP DESTIN	Yes			Yes
MD SOURCE	Yes	Yes	Yes	Yes
MD DESTIN	Yes			Yes
Ax DESTIN	Yes			Yes
Accumulator				
PV SOURCE			Yes	Yes

Related topics

“Reducing the number of data tables being scanned for Modbus controllers” on page 32

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.

Related topics

“Main properties for a Modbus controller” on page 28
“Serial communication settings for Modbus” on page 18

Serial communication settings for Modbus

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.

Related topics

"Main properties for a Modbus controller" on page 28
"Setting up a Modbus Plus network" on page 17

Modbus channel and controller reference

This section describes the configuration and addressing information specific to Modbus channels and controllers.

In addition to the information contained in this reference, and for help to build channels and controllers, see the section titled "Building controllers or channels" in the *Quick Builder User's Guide*.

Related topics

“Main properties for a Modbus channel” on page 20

“Port properties for a Modbus channel” on page 22

“Redundant port properties for a Modbus channel” on page 25

“Alternating behavior of redundant Modbus channels” on page 26

“Main properties for a Modbus controller” on page 28

“Optimizing Modbus scanning performance” on page 32

“Planning considerations for installing and configuring Modbus controllers” on page 5

Main properties for a Modbus channel

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

For information about how to create a channel, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

Property	Description
Name	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.
Description	(Optional) A description of the channel. A maximum of *132* alphanumeric characters, including spaces.
Marginal Alarm Limit	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. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. 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 *3* (which is the square root of 9) *x 25 = 75*.
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. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. 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. If there are multiple controllers configured on a channel, the diagnostic scan rate should be set to *60* seconds multiplied by the number of controllers on that channel. For example, if there are five controllers configured on the channel, the diagnostic scan rate should be set to *300* seconds. Diagnostic scans occur at the set scan rate per controller. Therefore, with five controllers and a default value of *60* seconds, the diagnostic scans will occur every *12* seconds.
Write Delay	If the channel is on a serial port, the length of time (in milliseconds) that the server waits before writing to any controller on the channel. The default value is *10* milliseconds. A write delay is usually specified only if: • The server communicates to the controller over a half-duplex radio link and the radio system requires time to key in each direction before the server or controller can send data. • The radio system implements *RTS/CTS* handshaking. If there is a communications problem and the controller does not respond to writes from the server, try changing this setting to *11* milliseconds or more. This should allow the controller enough time to become ready to receive data from the server.

Property	Description
Connect Timeout	The length of time that the server attempts to connect to the controller. The server will stop trying to connect to the controller once the timeout period passes. The default value *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 length of time that the server will wait for a reply from the controller. The server will stop waiting once the timeout period passes. 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	The type of channel specified when this item was created.
Last Modified	The date and time the channel properties were modified.
Last Downloaded	The date and time the channel was last downloaded to the server.
Item Number	The unique item number currently assigned to this channel, in the format *CHNcc*, where *cc* is the channel number. You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of channels allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Port properties for a Modbus channel

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 "Terminal server 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

Property	Description
Serial Port Name	The device name of the serial port.
Baud	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.
Number of Data Bits	Set this to *8*. This setting may be different if you are not using a 584 or 984 Modbus Controller.
Stop Bits	Set this to *1.* This setting may be different if you are not using a 584 or 984 Modbus Controller.
Parity	Set this to *EVEN*. This setting may be different if you are not using a 584 or 984 Modbus Controller.
Checksum	The type of checksum error detection used for the port. Not applicable for this channel. Select *NONE*.
XON/XOFF	The type of XON/XOFF software flow control used to stop a receiver from being overrun with messages from a sender. The types are: • *Input* (use XON/XOFF to control the flow of data on the receive line) • *None* (default) • *Output* (use XON/XOFF to control the flow of data on the transmit line) Set this to *None*.
RS-232	These options are applicable to the RS-232 link: • **Enable RTS/CTS flow control**. Select this if you want to use RTS/CTS for flow control to stop a receiver from being overrun with messages from a sender. • **Detect DCD**. 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. • **Detect DSR**. 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.

Property	Description
RS-485	These options are applicable to the RS-485 link: • **Enable Stallion RS-485 Half Duplex**. Select if RS-232 to RS-485 is done using a Stallion EasyConnection adapter. • **Echo (Required for Stallion RS-485 ports)**. Select so that the server expects the messages it sends to the port on the transmit line to be echoed back on the receive line. Select for a Stallion EasyConnection adapter or a Black Box converter.
Protocol	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*

Terminal server port properties

Property	Description
Terminal Server TCP Host Name Terminal Server TCP Port No	The name and port number of 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.
Idle Timeout	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.
Checksum	The type of checksum error detection used for the port. Not applicable for this channel. Select *NONE*.
Protocol	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*

LANVendor port properties

Property	Description
Adapter Number	(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
Network Type	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.

Specifying adapter number properties

Adapter Number Field Description	Corresponds to
Board0	Adapter 0 (Device 0)
Board1	Adapter 1 (Device 1)
Board2	Adapter 2 (Device 2)

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 "Redundant port properties for a Modbus channel" for more information on configuring redundant channels.

Related topics

“Redundant port properties for a Modbus channel” on page 25

Redundant port properties for a Modbus channel

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 section "Serial port properties" in the topic titled "Port properties for a Modbus channel" 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

“Port properties for a Modbus channel” on page 22

Alternating behavior of redundant Modbus channels

Alternating behavior of redundant controllers 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 controller to use the other link.

The **Disable Alternate Polling** check box on the controller's Main tab in Quick Builder enables and disables alternate polling. For more information about enabling and disabling alternate polling, see the topic titled "Main properties for a Modbus controller."

If Disable Alternate Polling is not selected

If the **Disable Alternate Polling** check box on the controller's Main tab is not selected (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 Disable Alternate Polling is selected

If the **Disable Alternate Polling** check box on the controller's Main tab is selected , 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

“Main properties for a Modbus controller” on page 28

Main properties for a Modbus controller

Use the Main tab to define the basic properties for a Modbus controller.

For information about how to create a controller, see the topic titled "Building controllers and channels" in the *Quick Builder User's Guide*.

Property	Description
Name	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.
Description	(Optional) A description of the controller. A maximum of *132* alphanumeric characters, including spaces.
Channel Name	The name of the channel on which the controller communicates with the server. (You must have already defined a channel for it to appear in this list.)
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A controller 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. The default value is *25*.
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the controller Marginal Alarm Limit. The default is *50*.
Dynamic Scanning Fastest Scan Period	Select the **Dynamic Scanning** check box to enable dynamic scanning of all point parameters on this controller. The default setting for this check box is selected. Define the fastest possible scan period (in seconds) that dynamic scanning will scan point parameters on this controller. The default is *15* seconds. The dynamic scanning period does not affect the static scanning rate for a parameter. For example, if the scanning rate for a parameter is 10 seconds, and the dynamic scanning rate for the controller is 15 seconds, the parameter will still be scanned at a period of 10 seconds.
Protocol	The protocol of the channel used by the controller.
Skip CR/LF	Shows whether the device skips the CR/LF characters at the end of each message.
PLC Station ID	ID of the controller, specified when configuring the controller—see the topics titled "Setting up a Modbus Plus network" and "Serial communication settings for Modbus" for more information.

Property	Description
Data Table	The data table that this controller represents. The address ranges for the data tables are as follows: • Digital Output (read/write). *000001* to *065535* • Digital Input (read only). *100001* to *165535* • Input Register (read only). *300001* to *365535* • Holding Register (read/write). *400001* to *465535* The addressable range for point parameter data is always 00001 to 32766 within each individual table. To access addresses in the range 32,767 to 65,535, you need to define a logical controller with an appropriate offset value. **Note**: When you enter an address, do not include the table number prefix.
Offset	The offset for the address range. *Offset + 1* is the first valid register. The offset must meet the following conditions: • *Offset* is less than the lowest register of the device that you want to read. • *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. The default value is *0*. Therefore, by default, the first valid register is *00001*, and the default register for the diagnostic scan is *00002*. 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." 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. **Attention** 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.
Diagnostic Address	(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. If specified, overrides the diagnostic address determined by the Offset value. See the section below titled "Diagnostic address examples" for more information. If this is not specified, the **Offset** value is read.
Use function code 16 to write Modbus device	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. Select this check box to use function code 16 to write data to registers within Modbus registers.
IP Address 1 and 2	(Applicable to Modbus TCP Protocol only) 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. If the IP Address is not specified, the Controller name is used as the TCP host name. For further information, see the Name property. 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.

Property	Description
Disable Alternate Polling	Alternating behavior of redundant channels is applicable only when the channel protocol for redundant channel is set to "Modbus TCP." If the **Disable Alternate Polling** check box is not selected (the default), requests will alternate between link A and B. If the **Disable Alternate Polling** check box is selected , then all requests will be made on the same link until the link fails.
Item Type	The type of controller specified when this item was created.
Last Modified	The date and time the controller properties were modified.
Last Downloaded	The date and time the controller was last downloaded to the server.
Item Number	The unique item number currently assigned to this controller, in the format *RTUnnnnn*. You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of controllers allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

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.

Offset	First hardware register	Diagnostic scan register
0	1	2
50	51	52
–30000	35536	35537

If you want to use an address range between 32,767 and 65,535, you must specify a negative offset and this value is calculated using the formula:

offset = 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 specific 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.

Offset	Diagnostic Address	Diagnostic scan register
0	0 or not specified	2
50	0 or not specified	52
0	20	21
50	20	71
–30000	0 or not specified	35537
–30000	200	35737

For diagnostic scan registers in the range between 32,767 and 65,535, the diagnostic scan register is calculated using one of the following formulas:

If the diagnostic address is greater than zero	If the diagnostic address is zero
diagnostic scan register = 65535 + offset + diagnostic address	*diagnostic scan register = 65537 + offset*

You can calculate the diagnostic address using the formula:

diagnostic address = diagnostic scan register – 65535 + 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: *offset = first hardware register to use – 65536*. Therefore, *offset = 40001 – 65536*, which is *–25525*.

To calculate the diagnostic address, use the formula: *diagnostic address = diagnostic scan register – 65535 – offset*. Therefore, *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

Attention
The calculations within this section are for negative offset values only.

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

address in Quick Builder = hardware register + 65535 - 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: *address in Quick Builder = hardware register + 65535 – offset*. Therefore, *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:

hardware register = 65535 + offset + address in Quick Builder

Related topics

“Function codes supported by Modbus” on page 14
“Alternating behavior of redundant Modbus channels” on page 26
“Serial communication settings for Modbus” on page 18
“Setting up a Modbus Plus network” on page 17

Optimizing Modbus 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 for Modbus controllers

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.

The scan packets that have been built can be listed by using the utility **lisscn** (list scan). Listing scan packets helps verify the scanning strategy.

For more information about **lisscn**, see the section titled "Command Reference" in the *Server and Client Configuration Guide*.

Reducing the number of data tables being scanned for Modbus controllers

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

“Data tables supported by Modbus” on page 15

Modbus points reference

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

In addition to the information contained in this reference, and for help to build points, see the section titled "Building and configuring points" in the *Quick Builder User's Guide*.

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

Related topics

“Defining a Modbus address for a point parameter” on page 34
“Planning considerations for installing and configuring Modbus controllers” on page 5

Defining a Modbus address for a point parameter

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

ControllerName Address

Part	Description
ControllerName	The name of the Modbus controller.
Address	The address within the controller where the value is stored. See the topic titled "Address syntax for Modbus controllers" for more information.

If you would like help with the address, you can use the Address Builder. To display the Address Builder, click ... next to **Address**.

Address syntax for Modbus controllers

The format for the address is:

Address [DataFormat|BitNumber]

Part	Description
Address	The address (decimal) for the parameter value within the controller table. If the address is prefixed with the table number, omit the table number. For example, the first register in table 4 is often written as 40001 or 400001. Enter *1* as the address. 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**.
DataFormat	The data format acronym or starting bit number, depending on how you want the value to be read: • Data format for scaling 16-bit integers. See the section below titled "Data format for scaling 16-bit integers." • Data format for reading floating point values. See the section below titled "Data format for reading floating point values." • Data format for reading raw values without scaling. See the section below titled "Data format for reading raw values without scaling." If you want to use a user-defined data format, you must define the format on the server. See the section titled "About user-defined data formats" in the *Server and Client Configuration Guide* for more information.
BitNumber	Only applicable to Input Register and Holding Register data tables. The starting bit number, ranging from *0* (default) to *15*, where *0* is the right-most bit in the 16-bit register. Analog and accumulator point parameters can read up to 16 bits. Status point parameters read 1, 2, or 3 consecutive bits.

Analog point

PV source: *161 U4095*

SP destination: *162 U4095*

Mode destination: *25 1*

Status point

PV source: *26*

Accumulator point

PV source: *171 C16*

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.

Data Format	Counts in Controller Register
U4095 (default)	0–4095
U9999	0–9999
U999	0–999
U15B	0–32767
U16B	0–65535
S16B	-32768–32767

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.

Data Format	Description	Counts in Controller Register
S32B	Signed 32-bit Binary Little-endian	-2147483648–2147483647
U32B	Unsigned 32-bit Binary Little-endian	0–4294967295
S32BS	Signed 32-bit Binary Scaled Little-endian	-2147483648–2147483647
U32BS	Unsigned 32-bit Binary Scaled Little-endian	0–4294967295

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.

Data Format	Description	Counts in Controller register
U32BB	Unsigned 32-bit Binary Big-endian	0–4294967295
U32BSB	Unsigned 32-bit Binary Scaled Big-endian	0–4294967295
S32BB	Signed 32-bit Binary Big-endian	-2147483648–2147483647

Data Format	Description	Counts in Controller register
S32BSB	Signed 32-bit Binary Scaled Big-endian	-2147483648–2147483647

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:

Format	Description
IEEEFPB	Bytes are big endian format (this is the same as IEEEFP)
IEEEFPBB	Bytes are byte-swapped big endian format
IEEEFPL	Bytes are little endian format
IEEEFPLB	Bytes are byte-swapped little endian format

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*.
- When reading floating point numbers with user defined data formats use two data addresses, **address** and **address +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).

Troubleshooting Modbus issues

This section describes troubleshooting tasks for Modbus that you can perform either on the server or from any Station.

Related topics

“Testing Modbus communications with the server” on page 38
“Troubleshooting Modbus point configuration errors” on page 39

Testing Modbus communications with the server

You use the Modbus test utility, **modtst**, to test communications between the server and the Modbus controller after you have downloaded channel and controller definitions to the server database.

Modbus controllers are fitted with transmit and receive diagnostic LEDs. If a communications fault illuminates an LED, see the Modbus manuals that shipped with your controllers.

Prerequisites

- Set up the controller.
- Connect all cables.
- Define the controller and channel in Quick Builder.
- Download the Quick Builder definitions to the server, without errors.
- Ensure the channel is out of service.

To run the modtst utility

1. Open a Command Prompt window.
2. Type **modtst** and then press Enter.
3. Follow the directions as prompted.
   You can read and write data to all registers that can be addressed by the server.

Related topics

“Planning considerations for installing and configuring Modbus controllers” on page 5

Troubleshooting Modbus 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®, PlantScape®, SafeBrowse®, TotalPlant®, and TDC 3000® are registered trademarks of Honeywell International, Inc.

OneWireless™ is a trademark 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.

Documentation feedback

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 Process Solutions Customer Contact Center (CCC) or 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 Process Solutions Customer Contact Center (CCC) or Honeywell Technical Assistance Center (TAC) listed in the “Support and other contacts” section of this document.

Support

For support, contact your local Honeywell Process Solutions Customer Contact Center (CCC). To find your local CCC visit the website, https://www.honeywellprocess.com/en-US/contact-us/customer-support-contacts/Pages/default.aspx.

Training classes

Honeywell holds technical training classes on Experion PKS. These classes are taught by experts in the field of process control systems. For more information about these classes, contact your Honeywell representative, or see http://www.automationcollege.com.

Index