

Experion
Controller References
Volume 3 of 3

EP-DSX114

300

11/05

Release 300

Document	Release	Issue	Date
EP-DSX114	300	0	November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] 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.

Support and other contacts

United States and Canada

Contact Honeywell IAC Solution Support Center
Phone 1-800 822-7673. In Arizona: (602) 313-5558
Calls are answered by dispatcher between 6:00 am and 4:00 pm Mountain Standard Time. Emergency calls outside normal working hours are received by an answering service and returned within one hour.
Facsimile (602) 313-5476
Mail Honeywell IS TAC, MS P13
2500 West Union Hills Drive
Phoenix, AZ, 85027

Europe

Contact Honeywell TAC-EMEA
Phone +32-2-728-2704
Facsimile +32-2-728-2696
Mail Honeywell TAC-EMEA
Avenue du Bourget, 1
B-1140 Brussels, Belgium

Pacific

Contact Honeywell Global TAC - Pacific
Phone 1300-300-4822 (toll free within Australia)
+61-8-9362-9559 (outside Australia)
Facsimile +61-8-9362-9169
Mail Honeywell Global TAC - Pacific
5 Kitchener Way
Burswood, WA, 6100, Australia
Email GTAC@honeywell.com

India

Contact Honeywell Global TAC - India
Phone +91-20-2682-2458 / 1600-44-5152
Facsimile +91-20-2687-8369
Mail Honeywell Automation India Ltd.
56 & 57, Hadapsar Industrial Estate
Hadapsar, Pune -411 013, India
Email Global-TAC-India@honeywell.com

Korea

Contact Honeywell Global TAC - Korea
Phone +82-2-799-6317
Facsimile +82-2-792-9015
Mail Honeywell Korea,
17F, Kikje Center B/D,
191, Hangangro-2Ga
Yongsan-gu, Seoul, 140-702, Korea
Email Global-TAC-Korea@honeywell.com

People's Republic of China

Contact Honeywell Global TAC - China
Phone +86-10-8458-3280 ext. 361
Mail Honeywell Tianjin Limited
17 B/F Eagle Plaza
26 Xiaoyhun Road
Chaoyang District
Beijing 100016, People's Republic of China
Email Global-TAC-China@honeywell.com

Singapore

Contact Honeywell Global TAC - South East Asia
Phone +65-6580-3500
Facsimile +65-6580-3501
+65-6445-3033
Mail Honeywell Private Limited
Honeywell Building
17, Changi Business Park Central 1
Singapore 486073
Email GTAC-SEA@honeywell.com

Taiwan

Contact Honeywell Global TAC - Taiwan
Phone +886-7-323-5900
Facsimile +886-7-323-5895
+886-7-322-6915
Mail Honeywell Taiwan Ltd.
10F-2/366, Po Ai First Rd.
Kaohsiung, Taiwan, ROC
Email Global-TAC-Taiwan@honeywell.com

Japan

Contact Honeywell Global TAC - Japan
Phone +81-3-5440-1303
Facsimile +81-3-5440-1430
Mail Honeywell K.K.
1-14-6 Shibaura Minato-Ku
Tokyo 105-0023
Japan
Email Global-TAC-JapanJA25@honeywell.com

Elsewhere

Call your nearest Honeywell office.

World Wide Web

To access Honeywell Solution Support Online, do the following:

- 1 In your web browser, type the address <http://www.honeywell.com/ps>.

- 2 Click **Login to My Account** and then log on.
- 3 Move the pointer over **Contacts & Support** in the top menu bar and then choose **Support** from the popup menu.

Contents

The Controller Reference Volumes 1, 2, and 3 contain separate “reference books” for each of the following controllers. Each book is entirely self-contained, with its own table of contents and index.

Volume 1	Volume 2	Volume 3
Allen-Bradley Interface Reference	Honeywell FSC Integration Reference	Honeywell UDC Integration Reference
Applicom Interface Reference	Honeywell LCS620 Interface Reference	Honeywell Universal Modbus Interface Reference
ASEA interface Reference	Honeywell Micromax Interface Reference	Modbus Interface Reference
Bristol Babcock Interface Reference	Honeywell RM7800 Flame Safeguard Interface Reference	Moore APACS Interface Reference
Bristol Babcock OpenBSI Interface Reference	Honeywell Series 9000 Integration Reference	Moore Mycro Interface Reference
DNP3 Protocol Interface Reference	Honeywell Safety Manager Integration Reference	OPC Client Interface
GE Fanuc Series 90 Interface Reference	Honeywell TDC 3000 Data Hiway Interface Reference	Siemens S5 and S7 Interface Reference
GEC GEM80 Interface Reference		Yamatake MA500 Interface Reference
Hitachi Interface Reference		
Honeywell DPR Interface Reference		
Honeywell EXCEL 5000 Interface Reference		

Honeywell UDC Integration Reference

Issue

4.1

Date

November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for UDC	6
	UDC models supported	6
	Other documentation	6
	UDC-specific terms	7
2	UDC controller setup	9
	Connecting the UDC to the server	10
	Configuring the communications settings	10
3	UDC controller configuration and addressing	11
	Defining a UDC channel	12
	UDC channel Main properties	12
	UDC channel Port properties	14
	Defining a UDC controller	16
	Loop address numbering	16
	UDC controller Main properties	17
	Defining a UDC address for a point parameter	19
	Entering an address	19
	UDC address definitions	19
	Optimizing scanning performance	25
	UDC scan packets	25
	Change the order of loop addressing	25
4	Server and Station tasks for UDC	27
	Testing server to UDC communications	28
	Using the hdwconfig utility	29
	Configuring the upload/download settings for UDC controllers	30
	Modifying the UDC configuration in Station	31
	Troubleshooting upload/download errors	31

Getting started

1

This reference describes how to set up, configure, and test Universal Digital Controllers (UDCs). There is detailed information for defining the channel and controller using Quick Builder.

Steps for connecting and configuring a UDC.

Complete each step before commencing the next step.

Step:	Go to:
Connect the UDC to the server	page 10
Set the communication parameters in the UDC	page 10
Define channels using Quick Builder	page 12
Define controllers using Quick Builder	page 16
Download channel and controller definitions to the server	
Test communications	page 28
Define controller points using Quick Builder	page 19

Support and documentation for UDC

UDC models supported

The server supports the following UDC models:

- UDC 3000
- UDC 5000
- UDC 6000

The server communicates with a UDC using RS-485 protocol. To use this protocol, the UDC must be fitted with the RS-485 communications option. Some models, manufactured prior to 1989, were not fitted with this option. For existing UDCs, we recommend that you upgrade to the latest firmware revision.

Other documentation

The following UDC documents are relevant for configuring the UDC:

- *UDC 3000 Product Manual* (51-52-25-07)
- *UDC 5000 Product Manual* (51-51-25-17)
- *UDC 6000 Process Controller Product Manual* (51-51-25-32)
- *UDC 3000/5000/6000 RS-422/485 Communications Option* (51-51-25-35)

The following documents for your server contain installation and configuration information:

- *Installation Guide*
- *Configuration Guide*

UDC-specific terms

Parameter Acronyms

Parameter acronyms are the server software names for the UDC parameters. They are an alternative to using the UDC parameter numbers.

UDC controller setup

2

This chapter describes how to set up a UDC.

The tasks for setting up a UDC are:

For:	Go to:
Connecting to the server	page 10
Communication settings	page 10

Connecting the UDC to the server

The UDC communicates with the server over an RS-485 link. Consequently, the UDC must be fitted with the RS-485 communications option.

For details about RS-485 links, see the *Installation Guide*. (This describes how to install and configure qualified RS-485 interfaces, such as Stallion.)

Configuring the communications settings

After connecting the UDC to the server, use the UDC's operator interface to configure the communication parameters.

- Communications options state (set to RS-422)
- Communication address for each loop (a unique number on the link from 1 to 99)
- Baud Rate (usually 9600)
- Parity (ODD)
- Duplex (Half)

The settings for Stop Bits (1) and Data Bits (7) are not configurable.



Note

Record these values because you need them when using Quick Builder—see “UDC controller configuration and addressing” on page 11.

UDC controller configuration and addressing

3

This chapter describes how to configure a UDC controller using Quick Builder.

For information about:	Go to:
Defining a UDC channel	page 12
Defining a UDC controller	page 16
Defining an address for a point parameter value	page 19
Optimizing scanning performance	page 25

Defining a UDC channel

You must define each physical communication link to the UDC network as a UDC channel.

To define a channel:

- 1 Click  to open the Add Items dialog box.
- 2 Select channel from **Add Items**.
- 3 Select UDC from **Type**.
- 4 Set the property values on the Main tab—see “UDC channel Main properties” on page 12.
- 5 Click the Port tab and for **Port Type** select either:
 - Serial
 - TerminalserverSee “UDC channel Port properties” on page 14.

UDC channel Main properties

The Main tab defines the basic properties for a channel.

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces, underscores or double quotes).
Description (optional)	A description of the channel. A maximum of 30 alphanumeric characters, including spaces.

Property	Description
Marginal Alarm Limit	The communications alarm limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. This limit is the channel barometer, which monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer is incremented by 2 or more, depending on the error and decremented for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel with the controllers' Marginal Alarm Limit (generally, you specify the same value for all controllers on a particular channel). For example, if there are 9 controllers, and you have set the controllers' Marginal Alarm Limit to 10, the value would be $\sqrt{9} \times 10$ (that is, 30).
Fail Alarm Limit	The communications alarm 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 in Marginal Alarm Limit.
Connect Timeout	The time, in seconds, the server attempts to connect to a controller before giving up. The default is 10 seconds.
Read Timeout	The time, in seconds, the server attempts to read data from a controller before giving up. The default value is 2 seconds.
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.
Item Number	The unique item number currently assigned to this item. 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.

UDC 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. See “Serial port properties” on page 14.
- **TerminalServer**. Select this if you want to connect the controller to the server via a LAN. See “TerminalServer port properties” on page 15.



Note

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

Serial port properties

The Serial Port properties are applicable to a Stallion EasyConnection adapter.

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	The number of data bits per second. The default is 9600.
Number of Data Bits	Set this to 7.
Stop Bits	Set this to 1.
Parity	Set this to ODD.
Checksum	Set this to None.
XON/XOFF	Set this to None.
RS-232	Not applicable. (The RS-232 and RS-485 settings are mutually exclusive.)
RS-485	Select Enable Stallion RS-485 Half Duplex and Echo . (Echo indicates that the server expects messages it sends to the port on the transmit line to be echoed back on the receive line.)

TerminalServer port properties

Property	Description
Terminal Server TCP Host Name	The name and port number of terminal server to which the channel is connected.
Terminal Server TCP Port No	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	Set to None.

Defining a UDC controller

The server requires a controller definition for each loop in a UDC. You need to define one controller for each UDC 3000, and up to two controllers for a UDC 3300, UDC 5000 or a UDC 6000 (these controllers can have two control loops).



Note

When defining a UDC 3300, UDC 5000 or UDC 6000 controller, ensure that you use an appropriate numbering scheme—see “Loop address numbering” on page 16.

To define a controller:

- 1 Click  to open the Add Items dialog box.
- 2 Select **Controller** from **Add Items**.
- 3 Select **UDC** from **Type**.
- 4 Set the property values on the **Main** tab—see “UDC controller Main properties” on page 17.

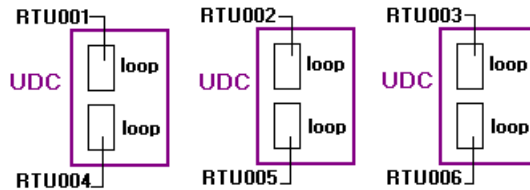
Loop address numbering

The server uses a separate controller to represent each of the 2 control loops in a single UDC 3300, UDC 5000 or UDC 6000.

The server sends a scan request to each controller consecutively by controller item number (loop number) in ascending order. Each UDC can only communicate a maximum of 3 times per second. If the 2 loops in a UDC have consecutive controller item numbers, up to 1/3 of a second can be wasted while the server waits for the UDC to accept a scan request for the second loop.

You avoid this problem, and spread the scan load efficiently, by interspersing loop numbering across UDCs so that the second loop in a particular controller is never scanned before the first loop has responded. The following figure shows optimal numbering for 3 UDCs, each with 2 control loops.

LOOP Numbering



UDC controller Main properties

The Main tab defines the basic properties for a controller.

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces, underscores or double quotes).
Description (optional)	A description of the controller. A maximum of 30 alphanumeric characters, including spaces.
Channel Name	The channel on which the controller communicates with the server.
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. This limit is the controller barometer, which monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer is incremented by 2 or more, depending on the error and decremented 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. Set this to double the value specified in Marginal Alarm Limit .
Controller Type	The UDC type this controller represents: either 3000, 5000 or 6000.

Property	Description
Loop Address Other Loop Address	The UDC's loop address(es), as defined when setting up the UDC. See "Configuring the communications settings" on page 10 and "Loop address numbering" on page 16. If the UDC has only one loop, leave Other Loop Address blank.
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 controller's property details.
Item Number	The unique item number currently assigned to this item. 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.

Defining a UDC address for a point parameter

Entering an address

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

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the UDC controller.
<i>Location</i>	The address within the controller where the value is stored. See “UDC address definitions” on page 19.

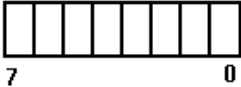
If you would like help when defining an address, click  next to **Address** to display Address Builder. For details, see the help.

UDC address definitions

The format for the location is:

Address [BitNumber]

Part	Description
<i>Address</i>	The parameter acronym that defines the UDC parameter. If the parameter value is PV, OP, SP and MD, use the LOOP acronym. The advantage of using the LOOP acronym is that the server acquires these parameter values in a single communications transaction with the UDC. For the list of UDC parameter acronyms, see: • “UDC 3000 parameter acronyms” on page 20 • “UDC 5000 parameter acronyms” on page 21 • “UDC 5000 parameter acronyms” on page 21

Part	Description
<i>BitNumber</i>	For UDC digital parameter values with codes between 128 and 255, enter the starting bit number for values contained in packed binary information. Bits are numbered from right to left as shown. Point parameters with a bit width greater than 1, reference successive bits from right to left. UDC Digital Parameter Bit Numbering  A diagram showing a horizontal row of eight empty square boxes. Below the boxes, the number '7' is centered under the first box on the left, and the number '0' is centered under the last box on the right, indicating bit numbering from left to right.

UDC 3000 parameter acronyms

You can use the following UDC 3000 acronyms when entering the source or destination controller address for a point parameter.



Note

For a complete list of UDC 3000 parameters and parameter numbers used by the server, see the file, `udc3000_def` in the server’s `\data` folder. If you need to use the parameter numbers for parameters not supported by the server software, you can find the full list in the UDC documentation.

Loop	Input1	Input1Bias
PV	Input2	Input1Filter
OP	Gain1	Burnout
CSP	Rate1	Frequency
LSP1	Reset1	Emissivity
LSP2	ManualReset	Input2Type
InternalRV	Gain2	Input2Char
OverrideSelect	Rate2	Input2High
Override1	Reset2	Input2Low
OPCalib0	CycleTime1	Input2Filter
OPCalib100	CycleTime2	PIDsets
SwType	SPSelect	InputFn2
SwVersion	AdapTune	LSPselect
ErrorStatus	Algo	OPrate
OptionStatus	DecPntLoc	Alarm1SP1Type

Input2install	TempUnits	Alarm1SP2Type
Input2option	Input1Type	Alarm2SP1Type
AdaptiveOption	Input1Char	Alarm2SP2Type
SetPointProg	Input1High	DigIn1
MathsOption	Input1Low	DigIn2
		DigInSta

UDC 5000 parameter acronyms

You can use the following UDC 5000 acronyms when entering the source or destination controller address for a point parameter.



Note

For a complete list of UDC 5000 parameters and parameter numbers used by the server, see the file, `udc5000_def` in the server's `\data` folder. If you need to use the parameter numbers for parameters not supported by the server software, you can find the full list in the UDC documentation.

Loop	DigInput1	Input1Burnout
PV	DigInput2	Input1Emis
OP	Input1	Input2Type
CSP	Input2	Input2Char
LSP1	Input3	Input2High
LSP2	Gain1	Input2Low
LSP3	Rate1	Input2Bias
InternalRV	Reset1	Input2Filter
Input1Override	ManualReset	Input2Burnout
Input2Override	Gain2	Input2Emis
PVOverride	Rate2	Input3Type
SPOVERRIDE	Reset2	Input3Char
Input3Override	CycleTime1	Input3High
Override1	CycleTime2	Input3Low
Override2	SPSelect	Input3Ratio
Override3	SPRamp	Input3Bias
SWType	ATune	Input3Filter
SWVersion	Algo	PIDsets

ErrorStatus	Loop2Sel	LSPselect
OptionStatus	Algo2	Dropoff
Input2install	InAlgo2	PIDsets2
Input3install	InAlgo3	LSPselect2
Input2option	Input1Type	Dropoff
Input3option	Input1Char	Alarm1Sp1Type
AutotuneOption	Input1High	Alarm1Sp2Type
MathsOption	Input1Low	Alarm2Sp1Type
SPPOption	InputBias	Alarm2Sp2Type
Loop2option	InputFilter	

UDC 6000 parameter acronyms

You can use the following UDC 6000 acronyms when entering the source or destination controller address for a point parameter.



Note

For a complete list of UDC 6000 parameters and parameter numbers used by the server, see the file, `udc6000_def` in the server's `\data` folder. If you need to use the parameter numbers for parameters not supported by the server software, you can find the full list in the UDC documentation.

Loop	Sched7gain	Gate5
PV	Sched8gain	Gate5InASta
OP	Sched1PV2	Gate5InAAna
CSP	Sched2PV2	Gate5InBSta
LSP1	Sched3PV2	Gate5InBAAna
LSP2	Sched4PV2	Gate5Out
LSP3	Sched5PV2	Input1Type
InternalRV	Sched6PV2	Input1Char
OverrideSelect	Sched7PV2	Input1High
Override1	Sched8PV2	Input1Low
Override2	Sched1gain2	Input1Ratio
Override2	Sched2gain2	Input1Bias
OPCalib0	Sched3gain2	Input1Filter
OPCalib100	Sched4gain2	Input2Type

SwType	Sched5gain2	Input2Char
SwVersion	Sched6gain2	Input2High
ErrorStatus	Sched7gain2	Input2Low
OptionStatus	Sched8gain2	Input2Ratio
Input2option	SPRamp	Input2Bias
Input3option	SPRate	Input2Filter
Input4option	AdapTune	Input3Type
Input5option	AdapTune2	Input3Char
AdaptiveOption	Algo	Input3High
MathsOption	Loop2Sel	Input3Low
Loop2option	Algo2	Input3Ratio
DigInput1	InAlgo1	Input3Bias
DigInput2	InAlgo2	Input3Filter
Input1	8SegChar	Input4Type
Input2	Polynomial	Input4Char
Input3	Totalizer	Input4High
Input4	Logic	Input4Low
Input5	Gate1	Input4Ratio
Gain1	Gate1InASta	Input4Bias
Rate1	Gate1InAAAna	Input4Filter
Reset1	Gate1InBSta	Input5Type
ManualReset	Gate1InBAAna	Input5Pulse
Gain2	Gate1Out	Input5Char
Rate2	Gate2	Input5High
Reset2	Gate2InASta	Input5Low
CycleTime1	Gate2InAAAna	Input5Ratio
CycleTime2	Gate2InBSta	Input5Bias
Sched1PV	Gate2InBAAna	Input5Filter
Sched2PV	Gate2Out	Input5BurnOut
Sched3PV	Gate3	Input5Emmis
Sched4PV	Gate3InASta	PIDsets
Sched5PV	Gate3InAAAna	LSPselect
Sched6PV	Gate3InBSta	OPrate
Sched7PV	Gate3InBAAna	PIDsets2
Sched8PV	Gate3Out	LSPselec2t

Sched1gain	Gate4	OPrate2
Sched2gain	Gate4InASta	Alarm1SP1Type
Sched3gain	Gate4InAAAna	Alarm1SP2Type
Sched4gain	Gate4InBSta	Alarm2Sp1Type
Sched5gain	Gate4InBAAna	Alarm2SP2Type
Sched6gain	Gate4Out	

Optimizing scanning performance

Minimize the number of scan packets—and the burden on the server—by using a small number of available scan periods for all your point definitions.

Example scan periods:

- 5 seconds for fast changing, important parameters;
- 60 seconds for 1 minute PV history, parameters that change every few minutes
- 300 seconds for slow changing parameters

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

UDC scan packets

When the **LOOP** parameter acronym is used as the UDC parameter value address, the **LOOP** parameter scans the PV, SP, OP, and MD value addresses in a single scan packet, thus optimizing scanning performance.

Two types of scan packets are created for UDCs:

- Periodic data acquisition
- Hardware diagnostic

A UDC can only process three scan packets per second. Therefore, never set the scan rate (scan period for the source address) to be faster than the UDC.

Change the order of loop addressing

Maximize processor response by ensuring that the controller item numbering is interspersed across several UDCs that have 2 loop addresses. See “Loop address numbering” on page 16.

Server and Station tasks for UDC

4

This chapter describes tasks for the UDC controller that you perform either on the server or from any Station.

For information about:	Go to:
Testing communications with the server	page 28
Saving the configuration image to a file	page 29
Configuring uploading/downloading settings	page 30
Modifying the configuration in Station	page 31

Testing server to UDC communications

You use the UDC diagnostic utility, **udctst**, to test communications, and to read and write individual UDC parameters.

See *UDC Product Manual* for details about UDC parameter numbers.



Caution

Do not run **udctst** while the server is communicating with UDCs—this may interfere with communications between them.

To run **udctst** on the server:

- 1 At the server command line, type **udctst**, and press ENTER.
You are prompted to “Enter LRN or device name of Channel”.
- 2 Type **chnxx**, where xx is the channel number you want to test.
You are prompted to enter a command.
The list of commands is also displayed. (If you want to display help about these commands, type **?** and press ENTER.)
- 3 To test the communication link, use the **F**ind command. This determines which UDC communication addresses are responding on the specified channel.
The communication addresses (that you set up on the UDCs and specified in your Quick Builder project file) should reply with a “responding” message. If this is not happening, ensure that:
 - Communication wiring is correct.
 - Communication parameters that you set up in the UDC, such as baud rate, agree with those you defined in Quick Builder.
 - The Stallion serial port (if used) has been set to use the RS-485 protocol.
 - Port is working by connecting to another serial device.
- 4 To quit, type **q** and press ENTER.



Note

When the server is sending controls (writes) to a UDC, it is not possible to configure that UDC by using the UDC operator interface display. This applies not only when testing communications, but also when scanning and configuring a UDC.

Using the hdwconfig utility

You use the configuration utility, **hdwconfig**, to save the server's configuration image of a UDC channel/controller to a file. (You can also restore the image from this file). Saving the configuration image to a file is useful for archiving purposes, as well as for transferring a configuration from one server to another.

Before saving, ensure that the server's configuration image reflects that of the UDC by performing an upload.

You run **hdwconfig** from a Command Prompt window.

To save channel or controller configuration data to a file, the syntax is:

```
HDWCONFIG SAVE file {-CHN cc| -RTU rr} [-RENEW]
```

To restore channel or controller configuration data from a file to the specified channel or controller with the same channel and controller, the syntax is:



Note

To use this command option, the system must be running and the controllers must be disabled.

```
HDWCONFIG RESTORE file {-CHN cc| cc-RTU rr} [-FROM nn]
```

To display a list of channels and controllers whose configuration data is contained in the file, the syntax is:

```
HDWCONFIG LIST file
```

Option	Description
SAVE <i>file</i>	Saves the configuration image to <i>file</i> .
RESTORE <i>file</i>	Restores configuration details from <i>file</i> .
LIST <i>file</i>	Displays a list of the channels and controllers contained in <i>file</i>
-CHN <i>cc</i>	Channel <i>cc</i>
-RTU <i>rr</i>	Controller <i>rr</i>
-RENEW	Saves the configuration from the specified file.
-FROM <i>nn</i>	Used in conjunction with -RTU <i>nn</i> to restore controllers from <i>nn</i> to <i>rr</i> .

For example, to save controller 21's configuration to a file called box21, type:

```
hdwconfig save box21 -rtu 21
```

Configuring the upload/download settings for UDC controllers

You use the Channel Configuration display to configure the upload/download settings for UDC controllers on a particular channel.

To configure upload/download settings:

- 1 In Station, select **Configure > Hardware > Channels**.
- 2 Double-click the channel you want to configure. The Channel Configuration display appears.

Item	Description
Upload permitted	If selected, uploading of all controllers on the channel is permitted.
Upload	If uploading is permitted, an operator clicks this button to start uploading. (An operator can also upload from a specific controller by clicking the Upload button on the System Configuration–Controllers display.)
Download permitted	If selected, downloading of controller on the channel database is permitted.
Download	If downloading is permitted, an operator clicks this button to start uploading. (An operator can also download from a specific controller by clicking the Download button on the System Configuration–Controllers display.)

Modifying the UDC configuration in Station

You can modify some UDC configuration settings from Station and then download them to the UDC.

To modify UDC configuration:

- 1 Select **Configure > Hardware > Controllers**. The Controllers listing appears.
- 2 Double-click the controller you want to configure. The Controller Configuration display appears.
- 3 Use the Group combo box at the top of the Controller Configuration display to look at the different groups of configuration information within a UDC. These groups are the same configuration groups which appear on the operator interface panel at the front of the UDC.
- 4 Change the parameter values as required. (Some parameters are read-only.) The server stores the new values in its configuration image. If the UDC is connected and in service, the server automatically sends them to the UDC. If a parameter value is not the same in both UDC and the server, it is shown as red in the display, rather than black. A discrepancy could be due to:

- An illegal value being passed to the UDC
- The channel not being configured to download
- The channel not being enabled
- The controller not being enabled
- A communication problem

Check the Message Zone for an explanation. You may need to perform an upload or download.

Troubleshooting upload/download errors

If errors occur while using Station to upload/download configuration data from the server to the controller, the parameter numbers of the affected parameter(s) are displayed in Station's Message Zone and are written to the log file.

These errors could be due to:

- Communications problems with the UDC
- Attempting to upload/download an incorrect configuration

Configuration parameter numbers associated with UDC configuration parameters are described in the following files in the server's \data folder:

- udc3000_def
- udc5000_def
- udc6000_def

Index

A

acronyms for parameters

UDC 3000 20

UDC 5000 21

UDC 6000 22

address definitions for a UDC 19

C

channel, defining 12

communication settings

UDC 10

communications, testing 28

config test utility 29

configuring a UDC in Station 31

configuring UDC, steps for 5

controller definition for a UDC controller 16

D

defining a UDC address for a point parameter 19

documentation

UDC related 6

documentation for UDC 6

download settings, configuring 30

E

errors, download 31

L

loop address numbering for a UDC 16

loop addressing, changing the order of 25

M

Main properties for a UDC channel 12

Main properties for a UDC Controller 17

models supported

UDC 6

P

parameter acronyms

UDC 3000 20

UDC 5000 21

UDC 6000 22

point parameter, defining an address 19

Port properties for a UDC 14

S

scanning

optimizing performance 25

packets 25

serial port properties for a UDC 14

server, connecting a UDC to 10

T

terminal server properties for a UDC 15

troubleshooting

download errors 31

U

UDC

communications settings 10

configuring in Station 31

connecting to a server 10

models supported 6

other documentation 6

upload settings, configuring 30

utility

config 29

Honeywell Universal Modbus Interface Reference

Issue

8.1

Date

November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for Modbus	6
	About the Modbus interface	6
	Other documentation for Modbus	6
	Control Products Wizard	8
2	Modbus controller setup	9
	Connecting your controller	10
	Setting up your connections	10
	Universal Modbus TCP connection.	14
	Configuring communication parameters.	16
	RS485 devices	16
	TCP/IP devices	16
3	Modbus controller configuration and addressing	17
	Defining a Modbus channel	18
	Modbus channel Main properties	18
	Modbus Port properties	20
	Defining a Modbus controller	24
	Modbus controller Main properties	24
	Defining a Modbus address for a point parameter value.	28
	Entering an address	28
	Named register addresses	34
	Troubleshooting point configuration.	35
	Errors when point building	35
	Error when scanning	35
	Optimizing scanning performance	36
4	Server and Station tasks for Modbus	37
	Testing communications with the server	38
	Troubleshooting Modbus configuration errors	40
	Common problems	40
	Problems with specific controller models	41
5	Device information	47
	Devices	49
	Baud rates supported	49

Numbered addresses 49

Non-numbered addresses 98

6 HC900 and UMC800 SPP and recipe support 101

Overview 102

Planning 103

 Resource requirements 103

 Migration/conversion requirements 104

Configuration 106

 Configuring a recipe 106

 Configuring a SP profile 107

 Configuring a combined recipe 108

 Configuring SPP monitoring 110

Operation 113

 Downloading a recipe 113

 Downloading a SP profile 114

 Downloading a combined recipe 114

 Controlling a SP programmer 115

Troubleshooting 117

Getting started

1

This reference describes how to set up, configure, and test Modbus controller communications with the server. There is detailed information for defining the controller using Quick Builder.

Complete each step before commencing the next.

Steps for Connecting and Configuring an Modbus Controller:

Steps:	Go to:
Connect and set up the Modbus controller according to the controller's user manual's instructions	page 10
Use Quick Builder to define channels	page 18
Use Quick Builder to define controllers	page 24
Download channel and controller definitions to the server	
Enable channels and test communications	page 38
Troubleshooting communication errors	page 40
Define points with Quick Builder	page 28

The Control Products wizard makes it easier to configure a controller if you have only used its standard options. See “Control Products Wizard” on page 8.

Support and documentation for Modbus

About the Modbus interface

The Modbus Interface enables the server to interface to any Control Products controller that implements the Honeywell Modbus protocol. The Honeywell Modbus protocol is the Honeywell implementation of the Modbus RTU Communications protocol for serial RS-485, RS-232 or Ethernet networks. Configuration information relating to specific controllers is supplied in separate user manuals (see “Other documentation for Modbus” on page 6).

This Interface is supported only by systems that are licensed for Modbus.

To check your system licence:

- 1 In Station, select **Configure > Server Licence Details** from the Station menu.
- 2 Go the **Interfaces** page.

All licensed options for your system display. Ensure that “Modbus” is listed.

Contact your local supplier for further licensing details.

Other documentation for Modbus

The following documents are available from Honeywell:

- *Modbus RTU Serial Communications User Manual* (Part number 51-52-25-66)
- *Overview and Planning Guide*
- *HC900 Modbus/TCP Communications User Manual* (Part number 51-52-25-111)

The controller communication and configuration user manuals are listed below.

Instrument Model	User Manual Part Number
RSX, VPR100, VRX100, UDC5300, VRX180	51-52-35-68
DR4300	51-52-25-71
DR4500	51-52-25-69
UDC 2300	51-52-25-75
UDC3300	51-52-25-70
	51-52-25-38 UDC3000 A Modbus 485 RTU Communication Manual

Instrument Model	User Manual Part Number
DPR180 / DPR250	EN11-6189 DPR180/DPR250 Communication Option Manual
UMC800	52-52-25-87 Modbus RTU Serial Communications User Manual
HC900	51-52-25-107 51-52-25-111 HC900 Hybrid Controller Communications User Guide
TrendView - Minitrend, Multitrend, ez Trend	43-TV-25-08 Communications Manual
Ethernet Bridge Card (UMC900, VRX180 / DPR180 / DPR250	51-52-25-96 Ethernet Interface Manual

Control Products Wizard

The Control Products wizard makes it easy to configure a controller and its associated points. The wizard guides you through the standard configuration tasks, and can download the configuration details to the server. You can also use the wizard to modify the configuration details of an existing controller.

You can access the wizard either in Quick Builder, where the wizard adds the controller to the current project. To start the wizard from Quick Builder, select it from the **Tools** menu. To use the wizard in Quick Builder, select:

Tools > Control Products Wizard.



Note

HC900 is not supported by the Control Products Wizard.

Modbus controller setup

2

This chapter describes how to set up an Modbus controller.

The tasks for setting up a Modbus controller are:

For:	Go to:
Connecting to the server	page 10
Setting Communications	page 16

Connecting your controller

Many different types of controllers can be connected to the same Modbus network. The only requirements are that every controller on the same network use the same connection type and baud rate, and that each controller use a unique device identification number.

Setting up your connections

Control Products controllers are designed to communicate using the Modbus TCP Ethernet, RS485 or the RS-232 specification. See the *User Manual* specific to your Control Products controller for information about cabling requirements.

Devices using RS-232 can be connected directly to a RS-232 port on the server, or to a terminal server.

Two methods are supported for connecting the server to an RS-485 network of Control Products controllers:

- Using an RS-232 to RS-485 converter (see “Using an RS-232/485 converter” on page 10).
- Directly connecting the server to the RS-485 network via an add-in card (see “Using an RS-485 adapter” on page 12).

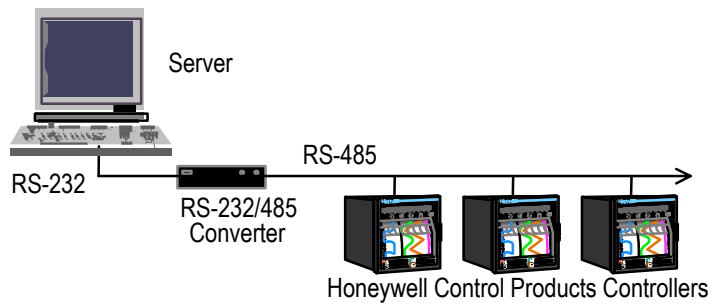
You can also connect to Ethernet TCP/IP networks using Modbus/TCP protocol using two methods:

- Direct Ethernet connection (HC900, TrendView)
- Ethernet - Modbus bridge (internal option for UMC800, VRX180, DPR180 and DPR250)

Make sure that you read the *User Manual* specific to your Control Products controller before connecting your controllers to the network.

Using an RS-232/485 converter

Honeywell recommends that you use the Black Box LD485-HS RS-232/485 Interface Converter, model number ME837A, or a Black Box IC109A-R2. These converters have been qualified by Honeywell. Use of another converter might produce unexpected results.

Figure 1 RS-232 to RS-485 converter

Connect an RS-232 port on the server to the RS-232 port on the Black Box converter using a standard RS-232 straight through cable. Then connect the Black Box converter and the Control Products controllers to the RS-485 network as shown in “Black Box (2-wire) connections” on page 11.

Black box connections

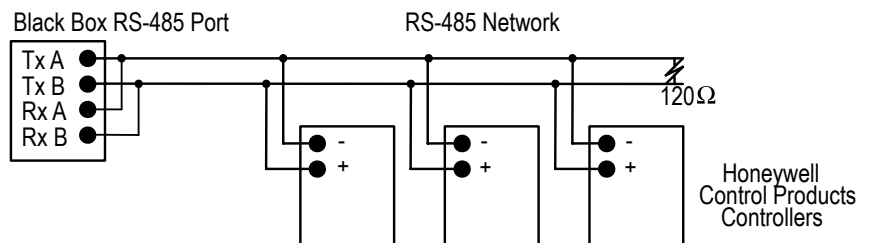
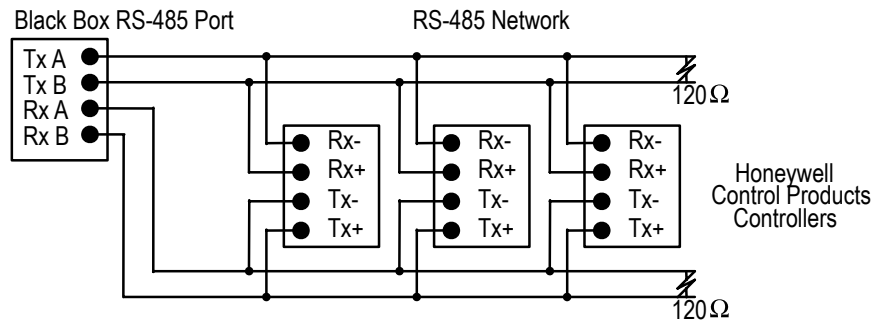
Figure 2 Black Box (2-wire) connections

Figure 3 Black Box (4-wire) Connections

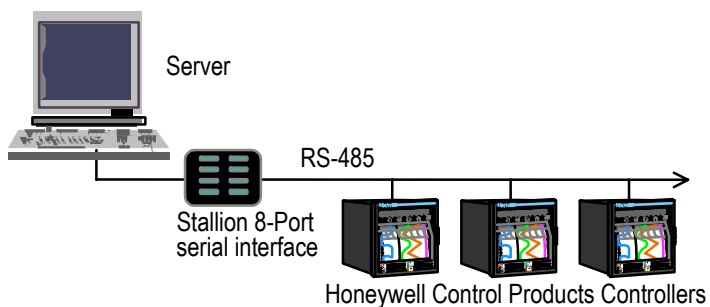


Ensure that the black box switches are configured with the following settings.

Switch	Setting	Description
XW1A	jumper in*	Configure RS-232 port as DCE.
XW1B	jumper out	Do not configure RS-232 port as DTE.
W8	B-C	2-wire (half-duplex) operation.
W9	C*	0 ms RTS/CTS delay.
W15	B-C	RS-485 transmitter enabled by data.
W5	A-B*	RTS/CTS normal.
W17	C	2 ms transmitter enabled time. This is good for 9600 Baud. Decrease for higher Baud rates. Increase for lower Baud rates. A - 30 ms B* - 7 ms C - 2 ms D - 0.7 ms E - 0.15 ms
W16	B*	0.1 ms delay before receiver enabled.
Term	ON	RS-485 receiver terminated.
Bias	OFF*	Line bias off.

Using an RS-485 adapter

Honeywell recommends using the Stallion EasyConnection 8/32 ISA, 8/32 PCI, 8/64 ISA or 8/64 PCI adapters with the Stallion RS-232 to RS-485 8-port dual interface asynchronous module. Honeywell has qualified this adapter. Use of another adapter may produce unexpected results.

Figure 4 Stallion EasyConnection Adapter

Installing the Stallion EasyConnection serial adapter

Stallion EasyConnection serial adapters are suitable for connection to RS-232, RS-422 and RS-485 devices.

Install the adapter, port module, and driver in the server as described in the Stallion documentation.

Connect a port on the Stallion port module directly to the RS-485 network as shown in "Stallion RS-485 (2-wire) connections" on page 13 and in "Stallion RS-485 (4-wire) connections" on page 14. Next, connect your Control Products controllers to the RS-485 network.

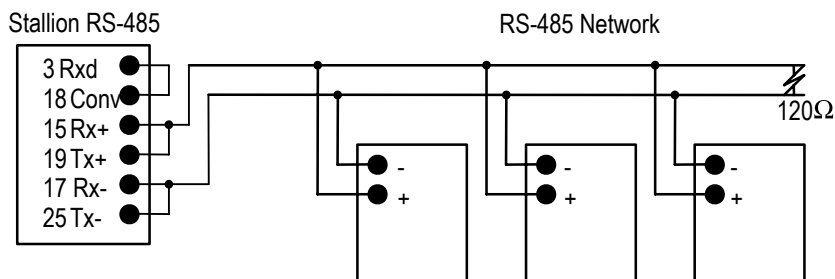
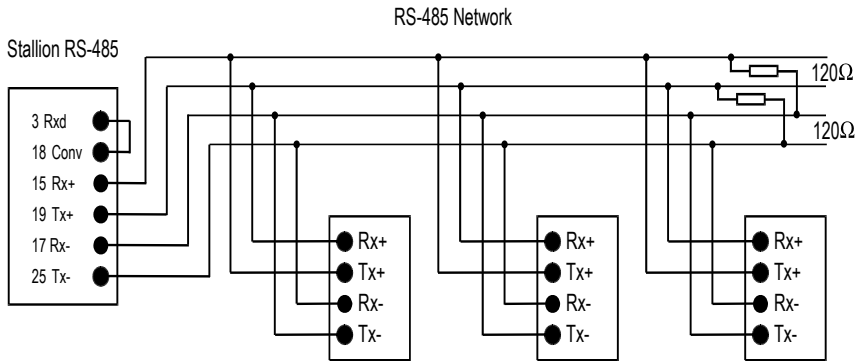
Figure 5 Stallion RS-485 (2-wire) connections

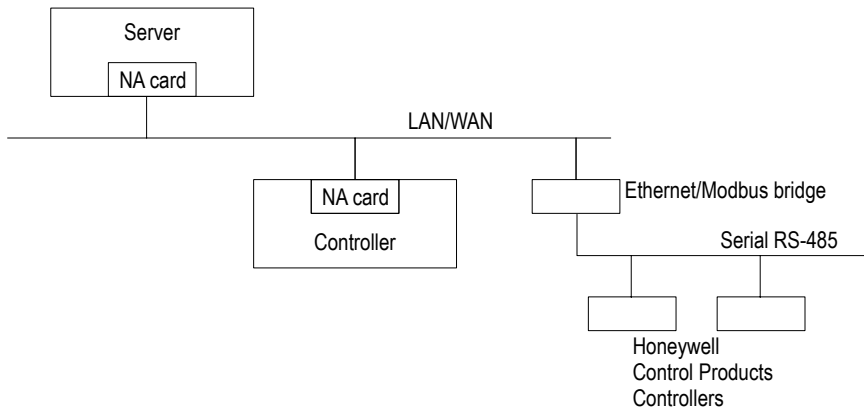
Figure 6 Stallion RS-485 (4-wire) connections



Universal Modbus TCP connection

To connect controllers to the server communicating using the UModbus 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. An external TCP/IP bridge (Lantronix DR1-IAP) may also be used for RS-485 network connection to Ethernet.

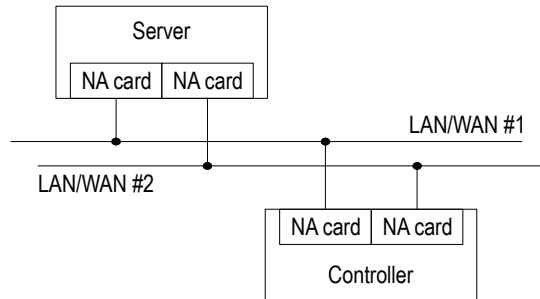
Figure 7 Non-redundant Universal Modbus TCP Network Architecture



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. This architecture is not presently supported in the controllers.

Figure 8 Redundant Universal Modbus TCP Network Architecture



Configuring communication parameters

RS485 devices

Before using your Control Products controllers, ensure that all communication parameters are configured correctly for each controller. Configure each controller to use the following communication parameters.

Parameter	Value
Number of Start Bits	1
Number of Data Bits	8
Number of Parity Bits	0
Number of Stop Bits	1

Make sure that each controller on the RS-485 network is configured for the same Baud rate. When you are ready to configure the server, you will need to know what Baud rate each RS-485 network is using.

Every controller using the same connection to the server (RS-232 or RS-485) should have a unique Modbus device identification number. Make a list showing what number has been associated with each of your controllers. You will need this information when using Quick Builder to configure the server to use your Control Products controllers.

TCP/IP devices

Ensure every controller or TCP/IP bridge device on the Ethernet network has a unique IP address. Make a list showing what IP addresses have been associated with each controller or bridge device. You will need this information when using Quick Builder to configure the server to use your Control Products controllers.

Note that any serial Control Products controllers connected to a TCP bridge must also conform to communications parameters for RS485 devices. Each must also have a unique physical address on the RS-485 network.

Modbus controller configuration and addressing

3

This chapter describes how to configure a Modbus controller for the system using Quick Builder. For each configuration procedure, there is detailed information covering all supported Modbus devices.

The Quick Builder controller configuration tasks are described:

For:	Go to:
Steps for defining a Modbus channel	page 18
Steps for defining a Modbus controller	page 24
Defining an address for a point parameter value	page 28
Defining a named address	page 28
Defining a non-named address	page 33
Troubleshooting point configuration	page 35
How to optimize scanning performance	page 36

You can simplify configuration tasks by using the Control Products wizard. See “Control Products Wizard” on page 8.

Defining a Modbus channel

Build a channel for each RS-485, RS-232 or Ethernet TCP/IP physical connection from your server. Note that you can multi-drop several controllers on the one serial RS-485 channel, provided they all use the same Baud settings.

To define a channel using Quick Builder:

- 1 Click  to add a channel.
- 2 In the Add Items dialog box, select Channel as the item and Modbus as the channel type.
- 3 Enter the channel details on the Main property page for the channel. For help with the channel definitions, see Modbus channel Main properties.
- 4 To complete the channel definition, click the Port tab and define either a serial, terminal server or a LANVendor port (for Ethernet Modbus/TCP). See “Modbus Port properties” on page 20.

Modbus channel Main properties

Use the Main tab to enter the basic channel properties for a Modbus channel.

Property	Description
Name	Type a unique name for the channel. A maximum of 10 alphanumeric characters can be used. (Double quotation marks and spaces are not allowed.)
Description	(Optional) Type a description of the channel. A maximum of 30 characters can be used, including spaces.

Property	Description
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. 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 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the average Marginal Alarm Limit defined for 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 $[\sqrt{9} \times 25 = 75]$.
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.
Write Delay	If the channel is on a serial port, specify the minimum number of milliseconds that the server must wait before writing to any controller on the channel. See “Channel write delay settings” on page 20.
Connect Timeout	Amount of time (in seconds) the server waits to connect to the controller before abandoning the connection. Type a new value in this field if you do not want the default, 10 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Read Timeout	Amount of time (in seconds) that the server waits for a reply from the controller. Type a new value in this field if you do not want the default, 2 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
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.

Property	Description
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

Channel write delay settings

Serial devices using the RS-485 protocol require a minimum period during which no communications occur. Different devices have different requirements. You should configure the write delay to be the largest value required by any device on your RS-485 network. See the following table for requirements of individual devices.

Where a delay is specified in number of characters, convert the value to milliseconds using this formula:

$$\text{Time(ms)} = (1000 \times \text{Characters}) / \text{Baud Rate}$$

Write delay should be rounded up to the nearest whole number.

For example, 3.5 Chars at 9600 Baud = $(1000 \times 3.5) / 9600 = 3.6\text{ms}$ (round to 4ms)

RSX, VPR, UDC5300, UMC800	DPR100, DPR180, DPR250	DR4300	DR4500	UDC3300, UDC2300
3.5 Chars	3.5 Chars	V 4: 20 ms V 5 or greater: 3.5 Chars + 2ms	V 57 and 58: 20ms V 59 or greater: 3.5 Chars + 2 ms	20 ms

Modbus Port properties

The Port tab defines the communication-related properties for a channel. The **Port Type** for Modbus can be:

- **Serial**. A serial communications interface, such as RS-485. See “Serial port properties” on page 21.
- **TerminalServer**. A communications link that enables controllers with a serial interface to be connected to a LAN. See “TerminalServer port properties” on page 22.
- **LANVendor**. A communications interface using TCP/IP to controllers connected to a LAN.

Serial port properties



Note

The Serial Port settings must match the settings on your communication devices.

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	The number of data bits per second. The default is 9600.
Number of Data Bits	The number of data bits used for transmission. The default is 8.
Stop Bits	The number of stop bits used for transmission. The default is 1.
Parity	The parity verification used on the port. The default is NONE .
Checksum	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: • None (default) • Input (use XON/XOFF to control the flow of data on the receive line) • Output (use XON/XOFF to control the flow of data on the transmit line)

Property	Description
Handshaking Options	RS-232 • Enable RTS/CTS flow control. Select to stop a receiver from being overrun with messages from a sender by using RTS/CTS for flow control. • Detect DCD. Select 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 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. Note: No options available for RS-422. RS-485 • Enable Stallion RS-485 Half Duplex. Select for a Stallion EasyConnection. • Echo (Required for RS-485 2-wire 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. Do not select for a Black Box Converter.

TerminalServer port properties

Property	Description
Terminal Server TCP Host Name	The name and port number of terminal server to which the channel is connected.
Terminal Server TCP Port No	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	Select None .

Redundant port properties

A communication port used as a redundant link has the same channel name but a requires a different port name to its twin. All other entries are identical to those of the primary port.

Defining a Modbus controller

You need to define a controller to describe each product on a channel.

To define a controller using Quick Builder:

- 1 Click  to add a controller.
- 2 In the Add Items dialog box, select Controller as the item and Modbus as the controller type.
- 3 Enter the property definitions for the controller on the controller Main property page.

Modbus controller Main properties

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

Property	Description
Name	Type a unique name for the controller. A maximum of 10 alphanumeric characters can be used. (Double quotation marks and spaces are not allowed.) For LAN connected controllers, the name must not contain Underscore (_) characters. This name is used to look up the IP address in the Hosts file or DNS 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) Type a meaningful description for this controller. A maximum of 30 characters, including spaces, can be entered.
Channel Name	The name of the channel on which the controller communicates. In the list of channel names, click the name. You need to have defined the channel in order for it's name to appear in the list.

Property	Description
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. Type a new value in this field if you do not want the default, 25.
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this value is reached, an urgent alarm is generated. Set this to double the value specified for the controller Marginal Alarm Limit.
Device Type	Enter the acronym for the type of controller you are using. See “Available device types” on page 26.
Device Identifier	The Modbus identification number assigned to your device.
Offset	Enter the lowest address within the range you intend to use. See “Using offsets” on page 26. By default use 0.
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.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field 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.
IP	If the channel Port Type is LANvendor, enter the controller’s IP address here. If the IP address is not specified, the controller name is used as the TCP host name. For more information see the Name property.

Available device types

Type Acronym	Controller Device
RSX	RSX Controller
VPR100	VPR100 Controller
VRX100	VRX100 Controller
VRX180	VRX180 Controller
UDC5300	Universal Digital Controller 5300
UDC2300	Universal Digital Controller 2300
UDC3300	Universal Digital Controller 3300
DR4300	DR4300 Circular Chart Recorder
DR4500	DR4500 Circular Chart Recorder
DPR180	Digital Process Recorder 180
DPR250	Digital Process Recorder 250
UMC800	UMC800 Controller
HC900	HC900 Controller
TV	TrendView Recorder

Using offsets

The server can only access a maximum of 4096 records in a particular file. Therefore if the server needs to access records beyond that limit, you may need to define several “logical” controllers in Quick Builder for a device, each with an appropriate offset.

For Modbus, use an offset to reference addresses outside the range 0x0000 and 0x1FFF. For example, if you have to refer to addresses between 0x0000 and 0x4000 within a device, you will need to create two controllers, one with an OFFSET=0 (the default) for all addresses up to 0x1FFF, and one with OFFSET=2000 for all addresses between 0x2000 and 0x3FFF.

HC900 and UMC800 controller OFFSET addresses

The Controller OFFSET address entry for the UMC800 and HC900 relative to parameter category is provided in the following table. For example, for an HC900, to access up to 24 control loops, all Variables, and up to 1000 Signal Tags would require setup of two virtual controllers with offset entries of 0 and 2000 respectively.

Parameter Category	OFFSET Address for Controller		Point Addressing
	UMC800	HC900	
Control Loops	0 (loops 1 to 16)	0 (loops 1 to 24)	Named (acronyms)
Control Loops (25 – 32)*	Not applicable	6000	Named (acronyms)
Variables (MATH_VAR)	0 (all Variables, 1-150)	0 (all Variables, 1-600)	Named (acronyms)
SP Programmers 1 to 4	0	0	Named (acronyms)
SP Programmers 5 to 8	Not applicable	Not supported	
Signal Tags (TAG)	2000 (Signal Tags 1–500)	2000 (Signal Tags 1–1000)	Named (acronyms)
Signal Tags 1001-2000	Not applicable	4000	Modbus Hex codes
SP Scheduler 1	2000	2000	Named (acronyms)
SP Scheduler 2	Not applicable	2000	Named (acronyms)
Sequencers 1 - 4	Not applicable	4000	Modbus Hex codes
Alternator, Stage, Ramp, HOA, Device Control	Not applicable	6000	Modbus Hex codes

* For Loop32, MODE is supported for Auto/Manual only.

Defining a Modbus address for a point parameter value

Different addresses are available depending on the type of device to which you are connected. Addresses that are *read-only* can only be used as source addresses. Addresses that are *write-only* can only be used as destination addresses. Addresses that are available for both read and write operations can be used as both source and destination addresses.

Entering an address

For source, and destination addresses the format for a Modbus controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the Modbus controller.
<i>Location</i>	The location in the controller where the value is recorded. The syntax depends on the address type: • “Location syntax for named addresses” on page 28 • “Location syntax for non-named addresses” on page 33

If you would like help when defining an address, click  next to **Address** to display Address Builder. For details, see the help.

Location syntax for named addresses

Named addresses can be either:

- Non-numbered Address
- Numbered Address

Non-numbered Address

For addresses that occur in only one location, specify the name of a register within your controller simply using the syntax:

AddressName [*Format*]

Part	Description
<i>AddressName</i>	Matches an address from “Non-numbered addresses” on page 98.

Part	Description
<i>Format</i>	(Optional) Specify only if the device does not use the default format for that address. Different addresses will have different default formats.

Numbered Address

For address types that occur multiple times within the device (for example, more than one analog input), use the syntax:

AddressName Number [SubAddressName] [Format]

Part	Description
<i>AddressName</i>	Name of the address, for example, loop. See “Numbered addresses” on page 49 for address names.
<i>Number</i>	The number of the address. See “Numbered addresses” on page 49 for address numbers.
<i>SubAddress Name</i>	(Optional) Some types of numbered addresses can have sub-addresses. For example, every loop has a Process Variable (PV) and a Set Point (WSP). See “Device information” on page 47.
<i>Format</i>	(Optional) Specify only if the device does not use the default format. See “Data formats” on page 33.

Example

The following example addresses the Process Variable (PV) of the second loop:

LOOP 2 PV

Process variable for loop 1:

LOOP 1 PV

Typical Control loop parameter addressing (where *n* is the loop number):

Parameter	Source Address	Destination Address
Process Variable (PV)	LOOP <i>n</i> PV	Not configurable
Set Point (SP)	LOOP <i>n</i> WSP*	LOOP <i>n</i> WSP
Output (OP)	LOOP <i>n</i> OPWORK**	LOOP <i>n</i> OPWORK
MODE (MD)	LOOP <i>n</i> LOOPSTAT	LOOP <i>n</i> MODEIN

* LSP1 or SP1 can be used if the parameter WSP is unavailable.

** OP can be used if the parameter OPWORK is unavailable.

Loop Tuning constants (possible AUX parameters for a loop point):

Parameter	Source Address	Destination Address
Gain	LOOP <i>n</i> GAIN1	LOOP <i>n</i> GAIN1
Reset	LOOP <i>n</i> RESET1	LOOP <i>n</i> RESET1
Rate	LOOP <i>n</i> RATE1	LOOP <i>n</i> RATE1

Digital Output values (used on a status point):

Parameter	Source Address	Destination Address
PV	DO <i>n</i>	Not configurable
OP	DO <i>n</i>	DO <i>n</i> *

* Note that not all devices support writing to digital output addresses. See “Digital output control strategies” on page 32.

Digit Input values (used on a status point):

Parameter	Source Address	Destination Address
PV	DI <i>n</i>	Not configurable

Extended loop support for HC900

For the HC900 controller, loops 25-32 must be addressed using the LOOPX parameter.

Parameter	Source Address	Destination Address
Process Variable (PV)	LOOPX <i>n</i> PV	Not configurable
Set Point (SP)	LOOPX <i>n</i> WSP	LOOPX <i>n</i> LSP1
Output (OP)	LOOPX <i>n</i> OPWORK	LOOPX <i>n</i> OPWORK
Mode (MD)	LOOPX <i>n</i> LOOPSTAT	LOOPX <i>n</i> MODEIN

Loop Tuning constants (possible AUX parameters for a loop point) for loops 25-32:

Parameter	Source Address	Destination Address
Gain	LOOPX <i>n</i> GAIN1	LOOPX <i>n</i> GAIN1
Reset	LOOPX <i>n</i> RESET1	LOOPX <i>n</i> RESET1
Rate	LOOPX <i>n</i> RATE1	LOOPX <i>n</i> RATE1

Signal tag and variable named address support for the HC900 and UMC800

Signal Tags (read only) with TAG as the named parameter and Variables (read/write) with MATH_VAR as the named parameter may be assigned to analog (floating point) or digital status points. The Variable and Signal Tag list (Tag Information) should be printed out from the controller configuration to obtain the sequential number listing and the data type (Analog or Digital) so that the proper point assignment may be made.

“Analog” Signal Tag Example:

Parameter	Source Address	Destination Address
PV	TAG <i>n</i>	Not Configurable

“Digital” Signal Tag Example:

Parameter	Source Address	Destination Address
PV	TAG <i>n</i>	Not Configurable

“Digital” Variable Example:

Parameter	Source Address	Destination Address
PV	MATH_VAR <i>n</i>	Not Configurable
OP	MATH_VAR <i>n</i>	MATH_VAR <i>n</i>

“Analog” Variable Example:

Parameter	Source Address	Destination Address
PV	MATH_VAR <i>n</i>	Not Configurable
SP	MATH_VAR <i>n</i>	MATH_VAR <i>n</i>

Digital output control strategies

Some controllers support the use of digital outputs as destination addresses; however, this functionality may have unintended consequences.

Digital outputs are typically controlled by the controller itself. If you use a digital output in a destination address, the server value will always override the value the controller expects to use. Once the output has been “forced” by the server, control cannot be returned to the controller. (that is, the server value will always have precedence).

Because of this potential problem, the use of the digital output as a destination address has been disabled for the UMC800. Instead if you have a control strategy as shown in “Digital Output Control Strategy - Example 1” on page 32, rather than use “Digital Output” as the destination of a server point parameter, use the strategy shown in the “Digital Output Control Strategy - Example 2” on page 32. This strategy uses two server destination addresses, “Force Value” and “Force Enabled”. “Force Enabled” enables you to switch between the local value, “Calculated Value”, and the server value, “Force Value”.

Figure 9 Digital Output Control Strategy - Example 1

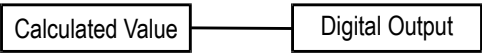
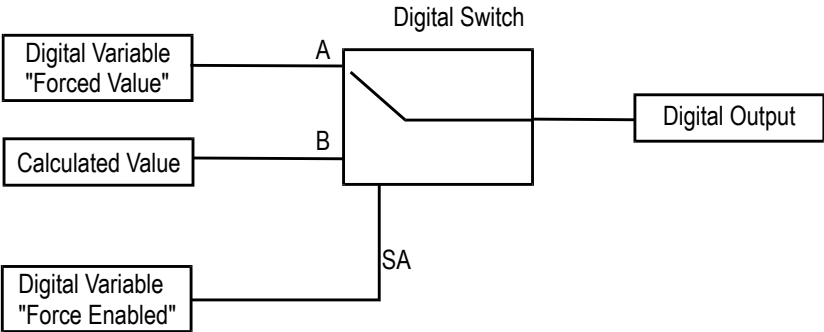


Figure 10 Digital Output Control Strategy - Example 2



Location syntax for non-named addresses

Addresses without names can be addressed directly using the format:

$n:0 \times A$ [*Format*]

Part	Description
n	Table number. See “Table types” on page 33 for table descriptions and their number.
A	Address within the table.
<i>Format</i>	(Optional) Only used for Input and Holding register tables (3 and 4). If a format is not specified, the format defaults to IEEEFP. See “Data formats” on page 33 for more information.

Table types

Table Description	Table Number	Point Type	Address Type
Digital Output (also known as Coil)*	0	Status	Source/Destination**
Digital Input	1	Status	Source
Input Registers	3	Status/Analog/Accumulator	Source
Holding Registers	4	Status/Analog/Accumulator	Source/Destination

* See “UMC800 problems” on page 41 and “HC900 problems” on page 43.

** See “Digital output control strategies” on page 32.

Data formats

The data format tells the server how to interpret the register value. The possible formats are:

Data Format	Description	Point Type
IEEEFP	32-bit IEEE floating point value. (Big Endian)	Status/Analog/Accumulator
n	Bit field. n represents the starting bit (0 to 15). This cannot be used with a named address.	Status
MODE	Informs the server that the address is a mode parameter.	Status/Analog/Accumulator
UINT2	Unscaled 16-bit integer.	Status/Analog/Accumulator

Non-named address examples (for HC900):

Parameter	Point Type	Address Type	Point Table/Address/ Format	Controller OFFSET Address
Signal Tag 1001	Analog	Source	4:0x42CD IEEEFP	4000
* Analog In, Slot 2, Channel 2, of Rack 2	Analog	Source	3:0x112 IEEEFP	0
* Digital In, Slot 8, Channel 3, of Rack 1	Status	Source	1:0x72	0
Step Number of Sequencer 1	Analog	Source	4:0x5AA9 UINT2 (16-bit Integer)	4000

* All I/O beyond Rack 1 in an HC900 must be addressed using hex addressing while all I/O in Rack 1 can be accessed using DI, DO, and AI named addresses, for example, D1 9, AI 18. See the HC900 Hybrid Controller Communications User Guide (part # 51-52-25-111) for a listing of I/O hex addresses.

Named register addresses

See “Device information” on page 47 for addresses relevant to your device.

Troubleshooting point configuration

Errors when point building

84E0h

Invalid address for this type of controller specified.

8426h

Invalid data format for this type of controller specified.

Error when scanning

(Errors appear in log)

0106h

A request to the controller timed out. This could be caused by a communication setup problems (for example, wrong address or Baud rate).

A channel write delay value being too low could also cause this problem. See “Channel write delay settings” on page 20. Try increasing the value to solve the problem.

8102h

An invalid address has been reported by the controller. This could be caused by the wrong address, the wrong data type, or the wrong controller type.

Optimizing scanning performance

The maximum amount of data that can be acquired from a controller is influenced by the rate of sending scan packets to the controller. An understanding of the Modbus scan packets will help you configure points so that optimal data acquisition performance can be achieved by maximizing the amount of data acquired with each scan packet.

The scan packets that have been built can be listed by using the list scan utility, **lisscn**. Listing scan packets helps verify the scanning strategy. See the *Configuration Guide* for usage of **lisscn**.

Server and Station tasks for Modbus

4

This chapter describes tasks for the Modbus controller that you perform either on the server or from any Station.

For:	Go to:
Testing communications with the server	page 38
Troubleshooting	page 40

Testing communications with the server

Use the test utility **umbtst** to test the communications.

Before using the utility, make sure that:

- You have set up your Control Products controllers according to their user manuals.
- All cables are connected.
- You have configured your channels in Quick Builder without error and downloaded all configuration information to the server without error.
- The server need not be running while using the utility as long as the database service is running. If making a connection through a terminal server, the server daemon service should also be running.
- The server is not communicating with your devices. The **umbtst** utility might interfere with communications.

To stop the server, type the command at the command prompt:

hscserver /load

Answer **y** to every prompt. This unloads the server, but leaves the database in memory.

To use the test utility, start a command prompt window and type: **umbtst**.

When prompted for the channel number, type **chn01** for channel 1 and so on.

For help using this utility, type **?**.

To check if your devices are present, use the **find a,b** command. This command locates all Modbus devices on the channel with IDs between a and b. For example:

```
C:\>umbtst
Enter LRN or device name of channel
chn01
Enter command:
find 1,4
FIND device with id 1 to 4, at 28-May-98 14:06:52

Device 1 ?
Device 2 ?
Device 3 ? ...responding
Device 4 ?
Enter command:
q
```


If you do not know the device name of your channel, select **View > System Status > Channels** from the **Station** menu. To the left of the channel name is the channel number. The device name of the channel will be the letters “chn” followed by the two-digit channel number. For example, your Modbus channel “COM3” might be channel number 1. Its device name will be “chn01”.

Troubleshooting Modbus configuration errors

Common problems

Error message/problem	Explanation/solution
You see the error in the server log file: Error code 0106 (Device Timeout)	The server has not received a response from the controller.
You see the error in the server log file: Error code 8102 (MODBUS error 2 - illegal data address)	You either specified an illegal address, or an illegal number of addresses.
You see the error in the Station Message Zone when you try to change the OP parameter: CONTROL - Illegal mode for control of parameter	The point is in AUTO mode, or its equivalents (AUTO-LSP, CASC, AUTO-RSP). You must change the mode of the point to MAN or its equivalents (MAN-LSP, MAN-RSP).
You see the error in the Quick Builder output file: Address is outside hardware cross reference table	You have upgraded your database from a previous server version and there is not enough room to store the controller addresses. To rectify the problem, follow these steps: 1. Make a backup of \server\data. 2. At the command prompt, type: sysbld -preserve -full. 3. Answer Y to the first two queries. 4. When presented with the ability to change all sorts of database values, press ENTER until you see the following message: There are 8192 addresses per rtu. Enter required number of addresses 5. Change the number of addresses per rtu (controller) to 32766. 6. Keep pressing ENTER until the sysbld command terminates.
The address LOOP <i>n</i> SP doesn't download.	The SP parameter is not a valid named address because there are a number of set point types available, and a simple SP is ambiguous. WSP stands for working set point and SP1 stands for set point 1. In most cases, WSP works best.

Problems with specific controller models

UDC3300 problems

Error message/problem	Explanation/solution
- You attempt to change a Setpoint in Station and the value changes on the Station display but not on the controller faceplate. - You attempt to change a Setpoint in Station and the value on the Station display changes to a different value and the controller's faceplate doesn't change.	The communications link between the controller and server can become overwhelmed. The solution is to increase the COM > Tx Delay on the controller faceplate.

UMC800 problems

Error message/problem	Explanation/solution
- You know that you should use the address LOOP <i>n</i> parametername but you don't know what value to use for <i>n</i>. - You want to access the process variable of the only PID loop you have configured. You used the number <i>n</i> which appears on the top right-hand corner of the PID block (LOOP <i>n</i> PV), but the values shown by the server don't seem to match those values in your controller.	The number shown on the top right-hand side of the PID block does NOT correspond with the loop number. You can find out the appropriate number by selecting File > Print > Block Parameters in the Honeywell Control Builder configuration utility. One of the properties printed out is Modbus® loop number. Use this number for your loops.
The address PID <i>n</i> PV doesn't download.	The PID part of the address is not valid and doesn't appear in the Modbus driver documentation. You cannot use the names of control blocks within Control Builder as Modbus addresses. You can only use the addresses listed in the Modbus documentation.

Error message/problem	Explanation/solution
You know that you should use the address AI n but you don't know what value to use for n .	The analog input number is calculated using the formula: $n = (m-1) * 4 + c$. n = the analog input number. m = the module/slot number. The UMC800 has 16 slots, numbered 1 to 16. c = the channel number (of the analog input). The analog input devices have up to four channels, numbered 1 to 4.
You know that you should use the address DI n or DO n but you don't know what value to use for n .	The digital input number is calculated using the formula: $n = (m-1) * 16 + c$. n = the digital input or output number. m = the module/slot number. The UMC800 has 16 slots, numbered 1 to 16. c = the channel number (of the digital input or output). The digital I/O devices have up to 16 channels, numbered 1 to 16.
You want to write to a digital output.	Honeywell recommends against writing to a digital output because this forces the output to a particular state, which cannot be overridden using the UMC800 internal logic. (Since this practice is inherently dangerous, it is not supported.) You can create a safer implementation using digital variables and some UMC800 logic blocks.
You see the error: ***** PNTBLD ERROR ***** illegal MODICON plc address in the Quick Builder output when trying to download a signal tag as a source address (such as TAG 2) to the server.	You might be trying to download to a controller whose OFFSET address is not 0x2000. Please read the Modbus documentation about address ranges and OFFSET.
You see the error: ***** PNTBLD ERROR ***** illegal MODICON plc address in the Quick Builder output when trying to download a signal tag, such as TAG 2, as a destination address to the server.	Signal tags are read-only parameters, so cannot be used as destination addresses. Please read the Modbus documentation about read-only and write-only addresses.

Error message/problem	Explanation/solution
You don't know what number to use for the signal tag using named address TAG or variable using named address MATH_VAR.	1. Start the configuration utility Honeywell Control Builder configuration utility. 2. Select File > Print. 3. Select Tag Properties then click OK. 4. Your printout should show, amongst other items, your signal tags. To the right of the words "Signal Tag" or "Variable" you should see a number. This is the tag number you should use in the address TAG <i>n</i> or MATH_VAR <i>n</i>.

HC900 problems

Error message/problem	Explanation/solution
You know you should use the address LOOP <i>n parametername</i> but you don't know what value to use for <i>n</i>. You want to access the process variable of the only control loop that you have configured. You used the number <i>n</i> which appears on the upper-right hand corner of the PID block (LOOP <i>n</i> PV for loops 1-24 and LOOPX <i>n</i> PV for loops 25-32), but the values shown by the server don't seem to match those values in your controller.	The block execution order number shown on the upper right-hand side of the block does NOT correspond with the loop number. The loop number corresponds with order of entry of the PID loop blocks only. You can find the appropriate number by selecting File > Print Report Preview, then select FBD's icon and Modbus Register Map > Summary Function Block Report in the Hybrid Control (HC) Designer configuration. The Loop Blocks are listed by number. Use this number for your loops.
For analog inputs in the first rack, you know that you should use the address AI <i>n</i> but you don't know what value to use for <i>n</i> .	For the first rack only, the analog input number is calculated using the formula: $n = (m-1) * 8 + c.$ <i>n</i> = the analog input number <i>m</i> = the module/slot number. The HC900 has up to 12 slots depending on rack size, numbered 1 to 12. <i>c</i> = the channel number (of the analog input). The analog input cards have 8 channels, numbered 1 to 8. The 2nd AI channel for slot/module 2 in Rack 1 is AI 10.

Error message/problem	Explanation/solution
I don't know to access analog inputs beyond the first rack.	If you have provided a Signal Tag for the Analog Input block output, use this tag number and TAG as the address name, for example, TAG 45 for a controller with an OFFSET address of 2000. Otherwise, you must use Non-Named hexadecimal addressing for a controller with an offset of 0. The address ranges for the racks are as follows: Rack 1: 0 - FF Rack 2: 100-1FF Rack 3: 200 – 2FF Rack 4: 300 – 3FF Rack 5: 400 – 4FF Zero-based addressing is used and two contiguous registers comprise the floating point data. Table 3 (Modbus Function Code 4) is used for access. The first analog channel for slot/module 1 in Rack 2 is: 3:x100 IEEFP, channel 2 is 3:x102 IEEFP, channel 8 is 3:x10E IEEFP. There are 8 inputs per slot/module.
For analog inputs in the first rack, you know that you should use the address DI n or DO n but you don't know what value to use for n .	For the first rack only, the digital or output number is calculated using the formula: $n = (m-1) * 16 + c.$ n = the analog input number m = the module/slot number. The HC900 has up to 12 slots depending on rack size, numbered 1 to 12. c = the channel number (of the digital input or output). The digital I/O cards have 8 or 16 channels, numbered 1 to 8 or 1 to 16. An allocation of 16 I/O is made for each slot/module regardless of type. The 2nd DI channel for slot/module 3 in Rack 1 is DI 34.

Error message/problem	Explanation/solution
I don't know to access digital I/O beyond the first rack.	If you have provided a Signal Tag for the Digital Input or Output block output, use this tag number and TAG as the address name, for example, TAG 56 for a controller with an OFFSET address of 2000. Otherwise, you must use Non-Named hexadecimal addressing for a controller with an offset of 0. The address ranges for the racks are as follows: Rack 1: 0 - FF Rack 2: 100-1FF Rack 3: 200 – 2FF Rack 4: 300 – 3FF Rack 5: 400 – 4FF Zero-based addressing is used and two contiguous registers comprise the floating point data. Table 1 is used for access to digital inputs and Table 0 is used for digital outputs. The 3rd digital input channel for slot/module 6 in Rack 2 is 1:x152, the 4th digital input on the same module is 1:x153. The 5th digital output for slot/module 8 in Rack 3 is 0:x274. There are 8 inputs per slot/module. An allocation of 16 I/O is made for each slot/module regardless of type.
You want to write to a digital output.	Honeywell recommends against writing to a digital output since this forces cannot be returned to normal via Modbus communications. Use the HC Designer tool concurrently for force actions where force removal is supported. You may also use digital Variables and logic blocks in the controller configuration to implement the force more safely via Station.
You see the error: ***** PNTBLD ERROR***** illegal MODICON plc address in the Quick Builder output when trying to download a signal tag as a source address (such as TAG 2) to the server.	You might be trying to download to a controller whose OFFSET address is not 2000. Please read the Universal Modbus documentation about offset ranges and OFFSET.

Error message/problem	Explanation/solution
You see the error: ***** PNTBLD ERROR***** illegal MODICON plc address in the Quick Builder output when trying to download a signal tag such as TAG 2 as a destination address to the server.	Signal tags are read-only parameters, so cannot be used as destination addresses. You will need to use Variables in your HC900 configuration instead for writes. Please read the Universal Modbus documentation about read-only and write-only addresses.
You don't know what number to use for accessing an HC900 Signal Tag or a Variable.	Start the Hybrid Control (HC) Designer configuration tool. Select File > Print Report Preview Select FBD's icon in the dialog box. Select Tag Information from the pull-down menu This listing shows the Variables and Signal Tags used in the configuration listed by tag name and in number sequence. Use the number in the # column as your reference for use in the address TAG <i>n</i> (for Signal Tags) or MATH_VAR <i>n</i> for Variables. You may print out this list for reference by selecting the Print button from Print Preview.
You want to know which HC900 Signal Tags or Variables are digital in nature so that they can be applied to Status points.	You can apply Signal Tags (read only) and Variables (read/write) to digital Status points if they are digital data types. See above for information related to viewing/printing the Tag Information Report. The Data Type column lists whether the parameter is Digital or Analog. If digital, you may apply to Status points. The UMB driver does the floating point conversion to integer translation to read or write an ON (1) or OFF (0) condition.
You want to know how to input a set point programmer point to use the standard screens in Station for viewing an HC900 set point programmer table and the profile pre-plot.	Consult the HC900 SPP & Recipe Support Users Guide. Support is for programmers 1-4 only. There is no UMB driver support for programmers 5-8.
You want to force a digital input on or off via status point OP parameter.	You cannot force a digital input except via the HC Designer configuration software (force DI block output in Monitor mode).

This chapter lists numbered and non-numbered addresses, their parameter details and the devices which are supported.

This following devices are supported by the Modbus controller:

- RSX
- VPR100
- VRX100
- VRX180
- UDC5300
- DR4300
- DR4500
- DPR180
- DPR250
- UDC2300
- UDC3300
- UMC800
- HC900
- TrendView

For numbered address details about:

For details about:	Go to:
Alarm set point value group	page 62
Alarm set point value group	page 63
Alarm status	page 65
Alarm status analog	page 66
Alarm status channel	page 67

For details about:	Go to:
Alarm status com	page 68
Alarm status digital	page 66
Alarm status event	page 67
Alarm status math	page 68
Analog input	page 58
Communication or constant value group	page 60
Digital input table	page 69
Digital input table	page 70
Digital output table	page 70
Digital output table	page 71
Math, variable or calculated value group	page 61
Math or calculated value status	page 64
PID loop	page 49
Set point scheduler #1 segment	page 91
Set point scheduler #2 segment	page 94
Set point scheduler values	page 87
Set point programmer	page 72
Set point programmer #1 profile segment	page 78
Set point programmer #2 profile segment	page 80
Set point programmer #3 profile segment	page 83
Set point programmer #4 profile segment	page 85
Set point program additional values	page 76
Tagged signal	page 59
Totalizer value group	page 62
Totalizer value status	page 64

For information about non-numbered addresses, see “Non-numbered addresses” on page 98.

Devices

Baud rates supported

The following table lists the devices and their supported baud rates.



Note

Baud rates are not applicable to HC900 or TrendView devices. These devices use Ethernet connections.

Device	Baud Rate Supported							
	300	600	1200	2400	4800	9600	19200	38400
RSX	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VPR100	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRX100	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
VRX180	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
UDC5300	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DR4300	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DR4500			Yes	Yes	Yes	Yes	Yes	
DPR180	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
DPR250	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
UDC2300	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
UDC3300	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
UMC800	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Numbered addresses

PID loop

The following table lists the devices which support PID Loops and their formats.

Device	Supported Address Format	Range
RSX	LOOP [n] [param]	[n] = 1 to 2
VPR100	LOOP [n] [param]	[n] = 1 to 4

Device	Supported Address Format	Range
VRX100	LOOP [n] [param]	[n] = 1 to 2
VRX180	LOOP [n] [param]	[n] = 1 to 8
UDC5300	LOOP [n] [param]	[n] = 1 to 2
DR4300	LOOP [n] [param]	[n] = 1 to 1
DR4500	LOOP [n] [param]	[n] = 1 to 2
UDC2300	LOOP [n] [param]	[n] = 1 to 1
UDC3300	LOOP [n] [param]	[n] = 1 to 2
UMC800	LOOP [n] [param]	[n] = 1 to 16
HC900	LOOP [n] [param]	[n] = 1 to 24
HC900	LOOPX [n] [param]	[n] = 25 to 32

The following table lists the details of the PID Loop parameters.

Param	Address Line	Param Format	Access	Devices
Process Variable	LOOP [n] PV *	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Output	LOOP [n] OP	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Input #1	LOOP [n] INP1	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Gain #1 (Prop Band #1 if active)	LOOP [n] GAIN1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Prop Band #1	LOOP [n] PROP1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
DIR	LOOP [n] DIR	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Reset #1	LOOP [n] RESET1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate #1	LOOP [n] RATE1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
PV Low Range	LOOP [n] PVLOW	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
PV High Range	LOOP [n] PVHIGH	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Alarm #1 SP #1	LOOP [n] AL1SP1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Alarm #1 SP #2	LOOP [n] AL1SP2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Gain #2 (Prop Band #2 if active)	LOOP [n] GAIN2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Prop Band #2	LOOP [n] PROP2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
DB	LOOP [n] DB	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Reset #2	LOOP [n] RESET2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate #2	LOOP [n] RATE2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point #1	LOOP [n] SP1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Local Set Point #1	LOOP [n] LSP1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Local Set Point #2	LOOP [n] LSP2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Alarm #2 SP #1	LOOP [n] AL2SP1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Alarm #2 SP #2	LOOP [n] AL2SP2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
SP Low Limit	LOOP [n] SPLOW	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
SP High Limit	LOOP [n] SPHIGH	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Output Low Limit	LOOP [n] OPLOW	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
OP High Limit	LOOP [n] OPHIGH	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Output Working Value	LOOP [n] OPWORK	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
RATIO	LOOP [n] RATIO	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
BIAS	LOOP [n] BIAS	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Deviation	LOOP [n] DEV	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Auto / Manual	LOOP [n] AMSTAT	Discrete (bits). [Status Point Only]	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Writable Controller Mode	LOOP [n] MODEIN	Control Mode - Auto / Man State (bit 0) and LSP / RSP State (bit 2).	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Remote / Local Set Point State	LOOP [n] RSP_STATE	Discrete (bits). [Status Point Only] Bit 0, 0 = LSP, 1 = RSP	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Tune Set State	LOOP [n] TUNE_SET_STATE	Discrete (bits). [Status Point Only]	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Read-only mode for the PID Loop	LOOP [n] LOOPSTAT	Mode Status - Bit 0 = Auto / Man State. Bit 2 = LSP / RSP State.	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Read-only AUTO / MAN Mode	LOOP [n] STATUS_MODE	Discrete (bits). [Status Point Only] Bit 0, 0 = Man, 1 = Auto	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Currently Selected Set Point	LOOP [n] STATUS_SP	Discrete (bits). [Status Point Only]	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Currently Selected Local or Remote Set Point	LOOP [n] STATUS_RSP	Discrete (bits). [Status Point Only]	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Currently Selected Tune Set	LOOP [n] STATUS_TUNE	Discrete (bits). [Status Point Only] Bit 0, 0 = Tune Set 1, 1 = Tune Set 2	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Remote Set Point (RSP)	LOOP [n] RSP	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point #2	LOOP [n] SP2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Working Set Point (WSP)	LOOP [n] SPWORK	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Remote Set Point (RSP)	LOOP [n] RSP	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point #2	LOOP [n] SP2	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Working Set Point (WSP)	LOOP [n] SPWORK	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300
Working Set Point (SPWORK)	LOOP [n] WSP	Floating Point	RO	DR4300, DR4500
Working Set Point (SPWORK)	LOOP [n] WSP	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, UDC2300, UDC3300, UMC800, HC900
Process Variable Override Value	LOOP [n] PVOVR	Floating Point	RW	UDC2300, UDC3300
Set Point Override Value	LOOP [n] SPOVR	Floating Point	RW	UDC2300, UDC3300
Output Override Value	LOOP [n] OPOVR	Floating Point	RW	UDC2300, UDC3300
Set Point State	LOOP [n] SP_STATE	Discrete (bits). [Status Point Only]	RW	UDC2300, UDC3300

Param	Address Line	Param Format	Access	Devices
Set Point State	LOOP [n] SP_STATE	Discrete (bits). [Status Point Only] Bit 0, 0 = SP1, 1 = SP2	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UMC800, HC900
Input #2	LOOP [n] INP2	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300
Cycle Time #1	LOOP [n] CYCLE1	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300
Cycle Time #2	LOOP [n] CYCLE2	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300
Local Set Point #3	LOOP [n] LSP3	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300
Temperature in carbon potential loop	LOOP [n] TEMP	Floating Point	RO	UMC800, HC900
Cycle Time #1	LOOP [n] CYCLE1	Floating Point	RO	UMC800, HC900
Cycle Time #2	LOOP [n] CYCLE2	Floating Point	RO	UMC800, HC900
Manual Reset	LOOP [n] MAN_RESET	Floating Point	RW	UMC800, HC900
Feed Forward Gain	LOOP [n] FF_GAIN	Floating Point	RW	UMC800, HC900
Local Percent Carbon Monoxide	LOOP [n] PCTCO	Floating Point	RW	UMC800, HC900
Furnace Factor	LOOP [n] FFCTR	Floating Point	RW	UMC800, HC900
Percent Hydrogen	LOOP [n] H2	Floating Point	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
On/Off Output Hysteresis	LOOP [n] OUT_ HYST	Floating Point	RW	UMC800, HC900
Carbon Potential Dewpoint	LOOP [n] CPD	Floating Point	RW	UMC800, HC900
Three Position Step Motor Time	LOOP [n] MOTOR	Floating Point	RW	UMC800, HC900
Fuzzy State	LOOP [n] FUZZY_ STATE	Discrete (bits). [Status Point Only] Bit 0, 0 = Disable, 1 = Enable	RW	UMC800, HC900
Demand Tune Request	LOOP [n] TUNE_ REQ	Discrete (bits). [Status Point Only] Bit 0, 0 = Off, 1 = On	RW	UMC800, HC900
Anti-soot set point limit enable	LOOP [n] ANTI_ SOOT	Discrete (bits). [Status Point Only] Bit 0, 0 = Off, 1 = On	RW	UMC800, HC900
IMAN Active / Inactive	LOOP [n] STATUS_ IMAN	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Active / Inactive LO	LOOP [n] STATUS_ LO	Discrete (bits). [Status Point Only]	RO	UMC800, HC900

* The default Parameter if only "LOOP [n]" is entered.

Analog input

The following table lists the devices which support Analog Inputs and their formats.

Device	Supported Address Format	Range
RSX	AI [n] [param]	[n] = 1 to 6
VPR100	AI [n] [param]	[n] = 1 to 12

Device	Supported Address Format	Range
VRX100	AI [n] [param]	[n] = 1 to 12
VRX180	AI [n] [param]	[n] = 1 to 48
UDC5300	AI [n] [param]	[n] = 1 to 3
DR4300	AI [n] [param]	[n] = 1 to 1
DR4500	AI [n] [param]	[n] = 1 to 4
DPR180	AI [n] [param]	[n] = 1 to 24
DPR250	AI [n] [param]	[n] = 1 to 64
UDC2300	AI [n] [param]	[n] = 1 to 2
UDC3300	AI [n] [param]	[n] = 1 to 3
UMC800	AI [n] [param]	[n] = 1 to 64
HC900	AI [n] [param]	[n] = 1 to 64*
TrendView	AI [n] [param]	[n] = 1 to 32

* In 1st rack, first 8 slots.

The following table lists the details of the Analog Input parameters.

Param	Address Line	Param Format	Access	Devices
Analog Input Value	AI [n] VALUE*	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, DPR180, DPR250, UDC2300, UDC3300, UMC800, HC900, TV

* The default Parameter if only "AI [n]" is entered.

Tagged signal

The following table lists the devices which support Tagged Signal and their formats.

Device	Supported Address Format	Range
UMC800	TAG [n] [param]	[n] = 1 to 500
HC900	TAG [n] [param]	[n] = 1 to 1000

The following table lists the details of the Tagged Signal parameters.

Param	Address Line	Param Format	Access	Devices
Tagged Signal Value	TAG [n] VALUE*	Floating Point	RO	UMC800, HC900

* The default Parameter if only “TAG [n]” is entered.

Communication or constant value group

The following table lists the devices which support the Communication or Constant Value group and their formats.

Device	Supported Address Format	Range
RSX	CN [n] [param]	[n] = 1 to 10
VPR100	CN [n] [param]	[n] = 1 to 16
VRX100	CN [n] [param]	[n] = 1 to 16
VRX180	CN [n] [param]	[n] = 1 to 32
UDC5300	CN [n] [param]	[n] = 1 to 9
DPR180	CN [n] [param]	[n] = 1 to 24
DPR250	CN [n] [param]	[n] = 1 to 32
TrendView	CN [n] [param]	[n] = 1 to 32

The following table lists the details of the Communication or Constant Value Group parameters.

Param	Address Line	Param Format	Access	Devices
Communication Value	CN [n] VALUE*	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DPR180, DPR250, TV**

* The default Parameter if only “CN [n]” is entered.

** TrendView communication values are written via this parameter, read via Math/Calculated values parameter (the “pen” value).

Math, variable or calculated value group

The following table lists the devices which support the Math or Calculated Value group and their formats.

Device	Supported Address Format	Range
RSX	MATH_VAR [n] [param]	[n] = 1 to 24
VPR100	MATH_VAR [n] [param]	[n] = 1 to 32
VRX100	MATH_VAR [n] [param]	[n] = 1 to 32
VRX180	MATH_VAR [n] [param]	[n] = 1 to 64
UDC5300	MATH_VAR [n] [param]	[n] = 1 to 16
DR4500	MATH_VAR [n] [param]	[n] = 1 to 1
DPR180	MATH_VAR [n] [param]	[n] = 1 to 24
DPR250	MATH_VAR [n] [param]	[n] = 1 to 32
UDC3300	MATH_VAR [n] [param]	[n] = 1 to 2
UMC800	MATH_VAR [n] [param]	[n] = 1 to 150
HC900	MATH_VAR [n] [param]	[n] = 1 to 600
TrendView	MATH_VAR [n] [param]	[n] = 1 to 64

The following table lists the details of the Math or Calculated Value Group parameters.

Param	Address Line	Param Format	Access	Devices
Math or Calculated Value	MATH_VAR [n] VALUE*	Floating Point	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4500, DPR180, DPR250, UDC3300, TV**
Math or Calculated Value	MATH_VAR [n] VALUE*	Floating Point	RW	UMC800***, HC900***

* The default Parameter if only "MATH_VAR [n]" is entered.

** Applies to TrendView "pen" values.

*** Applies to UMC800, HC900 "variable" values.

Totalizer value group

The following table lists the devices which support the Totalizer Value group and their formats.

Device	Supported Address Format	Range
RSX	TOTALIZER [n] [param]	[n] = 1 to 6
VPR100	TOTALIZER [n] [param]	[n] = 1 to 3
VRX100	TOTALIZER [n] [param]	[n] = 1 to 12
VRX180	TOTALIZER [n] [param]	[n] = 1 to 48
DR4300	TOTALIZER [n] [param]	[n] = 1 to 1
DR4500	TOTALIZER [n] [param]	[n] = 1 to 4
UDC3300	TOTALIZER [n] [param]	[n] = 1 to 1
TrendView	TOTALIZER [n] [param]	[n] = 1 to 64

The following table lists the details of the Totalizer Value Group parameters.

Param	Address Line	Param Format	Access	Devices
Totalizer Value	TOTALIZER [n] VALUE*	Floating Point	RO	RSX, VPR100, VRX100, VRX180
Totalizer Value	TOTALIZER [n] VALUE*	Floating Point	RW	DR4300, DR4500, UDC3300

* The default Parameter if only "TOTALIZER [n]" is entered.

Alarm set point value group

The following table lists the devices which support the Alarm Set Point Value group and their formats.

Device	Supported Address Format	Range
RSX	ALMSP [n] [param]	[n] = 1 to 12
VPR100	ALMSP [n] [param]	[n] = 1 to 16
VRX100	ALMSP [n] [param]	[n] = 1 to 16
VRX180	ALMSP [n] [param]	[n] = 1 to 96
UDC5300	ALMSP [n] [param]	[n] = 1 to 4
DPR180	ALMSP [n] [param]	[n] = 1 to 48
DPR250	ALMSP [n] [param]	[n] = 1 to 64

The following table lists the details of the Alarm Set Point Value Group parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Set Point Value	ALMSP [n] VALUE*	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DPR180, DPR250

* The default Parameter if only "ALMSP [n]" is entered.

Alarm set point value group

The following table lists the devices which support the Alarm Set Point Value group and their formats.

Device	Supported Address Format	Range
DR4300	ALMSP [n] [param]	[n] = 1 to 2
DR4500	ALMSP [n] [param]	[n] = 1 to 6
UDC2300	ALMSP [n] [param]	[n] = 1 to 2
UDC3300	ALMSP [n] [param]	[n] = 1 to 2

The following table lists the details of the Alarm Set Point Value Group parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Set Point #1	ALMSP [n] SP1*	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300
Alarm Set Point #2	ALMSP [n] SP2*	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300

* The default Parameter if only "ALMSP [n]" is entered.

Math or calculated value status

The following table lists the devices which support the Math or Calculated Value Status and their formats.

Device	Supported Address Format	Range
RSX	MATH_STATUS [n] [param]	[n] = 1 to 24
VPR100	MATH_STATUS [n] [param]	[n] = 1 to 32
VRX100	MATH_STATUS [n] [param]	[n] = 1 to 32
VRX180	MATH_STATUS [n] [param]	[n] = 1 to 64
UDC5300	MATH_STATUS [n] [param]	[n] = 1 to 16
DR4500	MATH_STATUS [n] [param]	[n] = 1 to 1
DPR180	MATH_STATUS [n] [param]	[n] = 1 to 24
DPR250	MATH_STATUS [n] [param]	[n] = 1 to 32
UDC3300	MATH_STATUS [n] [param]	[n] = 1 to 2

The following table lists the details of the Math or Calculated Value Status parameters.

Param	Address Line	Param Format	Access	Devices
Math or Calculated Value Status	MATH_STATUS [n] STATUS*	Discrete (bits). [Status Point Only]	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4500, DPR180, DPR250, UDC3300

* The default Parameter if only "MATH_STATUS [n]" is entered.

Totalizer value status

The following table lists the devices which support the Totalizer Value Status and their formats.

Device	Supported Address Format	Range
RSX	TOTALIZER_STATUS [n] [param]	[n] = 1 to 6
VPR100	TOTALIZER_STATUS [n] [param]	[n] = 1 to 3
VRX100	TOTALIZER_STATUS [n] [param]	[n] = 1 to 12
VRX180	TOTALIZER_STATUS [n] [param]	[n] = 1 to 48
DR4300	TOTALIZER_STATUS [n] [param]	[n] = 1 to 1

Device	Supported Address Format	Range
DR4500	TOTALIZER_STATUS [n] [param]	[n] = 1 to 4
UDC3300	TOTALIZER_STATUS [n] [param]	[n] = 1 to 1

The following table lists the details of the Totalizer Value Status parameters.

Param	Address Line	Param Format	Access	Devices
Totalizer Status	TOTALIZER_STATUS [n] STATUS*	Discrete (bits). [Status Point Only] 0 = Totalizer Off 1 = Totalizer On	RO	RSX, VPR100, VRX100, VRX180, DR4300, DR4500, UDC3300

* The default Parameter if only "TOTALIZER_STATUS [n]" is entered.

Alarm status

The following table lists the devices which support the Alarm Status and their formats.

Device	Supported Address Format	Range
RSX	ALMSTAT [n] [param]	[n] = 1 to 12
VPR100	ALMSTAT [n] [param]	[n] = 1 to 16
VRX100	ALMSTAT [n] [param]	[n] = 1 to 16
VRX180	ALMSTAT [n] [param]	[n] = 1 to 96
UDC5300	ALMSTAT [n] [param]	[n] = 1 to 4
DR4300	ALMSTAT [n] [param]	[n] = 1 to 2
DR4500	ALMSTAT [n] [param]	[n] = 1 to 6
UDC2300	ALMSTAT [n] [param]	[n] = 1 to 2
UDC3300	ALMSTAT [n] [param]	[n] = 1 to 2
UMC800	ALMSTAT [n] [param]	[n] = 1 to 120
HC900	ALMSTAT [n] [param]	[n] = 1 to 120

The following table lists the details of the Alarm Status parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT [n] STATUS*	Discrete (bits). [Status Point Only]	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

* The default Parameter if only "ALMSTAT [n]" is entered.

Alarm status analog

The following table lists the devices which support the Alarm Status Analog and their formats.

Device	Supported Address Format	Range
DPR180	ALMSTAT_ANALOG [n] [param]	[n] = 1 to 24
DPR250	ALMSTAT_ANALOG [n] [param]	[n] = 1 to 64

The following table lists the details of the Alarm Status Analog parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT_ANALOG [n] STATUS*	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

* The default Parameter if only "ALMSTAT_ANALOG [n]" is entered.

Alarm status digital

The following table lists the devices which support the Alarm Status Digital and their formats.

Device	Supported Address Format	Range
DPR180	ALMSTAT_DIGITAL [n] [param]	[n] = 1 to 36
DPR250	ALMSTAT_DIGITAL [n] [param]	[n] = 1 to 48

The following table lists the details of the Alarm Status Digital parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT_ DIGITAL [n] STATUS*	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

* The default Parameter if only "ALMSTAT_DIGITAL [n]" is entered.

Alarm status event

The following table lists the devices which support the Math or Calculated Value Status and their formats.

Device	Supported Address Format	Range
DPR180	ALMSTAT_EVENT [n] [param]	[n] = 1 to 6
DPR250	ALMSTAT_EVENT [n] [param]	[n] = 1 to 6

The following table lists the details of the Alarm Status Event parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT_EVENT [n] STATUS*	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

* The default Parameter if only "ALMSTAT_EVENT [n]" is entered.

Alarm status channel

The following table lists the devices which support the Alarm Status Channel and their formats.

Device	Supported Address Format	Range
DPR180	ALMSTAT_CHANNEL [n] [param]	[n] = 1 to 24
DPR250	ALMSTAT_CHANNEL [n] [param]	[n] = 1 to 64

The following table lists the details of the Alarm Status Channel parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT_CHANNEL [n] STATUS*	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

* The default Parameter if only "ALMSTAT_CHANNEL [n]" is entered.

Alarm status com

The following table lists the devices which support the Alarm Status Com and their formats.

Device	Supported Address Format	Range
DPR180	ALMSTAT_COM [n] [param]	[n] = 1 to 24
DPR250	ALMSTAT_COM [n] [param]	[n] = 1 to 32

The following table lists the details of the Alarm Status Com parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT_COM [n] STATUS*	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

* The default Parameter if only "ALMSTAT_COM [n]" is entered.

Alarm status math

The following table lists the devices which support the Alarm Status Math and their formats.

Device	Supported Address Format	Range
DPR180	ALMSTAT_MATH [n] [param]	[n] = 1 to 24
DPR250	ALMSTAT_MATH [n] [param]	[n] = 1 to 32

The following table lists the details of the Alarm Status Math parameters.

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT_MATH [n] STATUS*	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

* The default Parameter if only "ALMSTAT_MATH [n]" is entered.

Digital input table

The following table lists the devices which support the Digital Input Table and their formats.

Device	Supported Address Format	Range
RSX	DI [n] [param]	[n] = 1 to 6
VPR100	DI [n] [param]	[n] = 1 to 24
VRX100	DI [n] [param]	[n] = 1 to 24
VRX180	DI [n] [param]	[n] = 1 to 36
UDC5300	DI [n] [param]	[n] = 1 to 3
DR4300	DI [n] [param]	[n] = 1 to 2
DR4500	DI [n] [param]	[n] = 1 to 2
UDC3300	DI [n] [param]	[n] = 1 to 2
UMC800	DI [n] [param]	[n] = 1 to 256
HC900	DI [n] [param]	[n] = 1 to 256
TrendView	DI [n] [param]	[n] = 1 to 32

The following table lists the details of the Digital Input Table parameters.

Param	Address Line	Param Format	Access	Devices
Digital Input Value	DI [n] VALUE*	Discrete (bits). [Status Point Only]	RO	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC3300, UMC800, HC900**, TV

* The default Parameter if only "DI [n]" is entered.

** Applies to Rack 1 only, allocation is 16 inputs (bits) per slot, 12 slots maximum.

Digital input table

The following table lists the devices which support the Digital Input Table and their formats.

Device	Supported Address Format	Range
DPR180	DI [n] [param]	[n] = 1 to 36
DPR250	DI [n] [param]	[n] = 1 to 48

The following table lists the details of the Digital Input Table parameters.

Param	Address Line	Param Format	Access	Devices
Digital Input Value	DI [n] VALUE*	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

* The default Parameter if only "DI [n]" is entered.

Digital output table

The following table lists the devices which support the Digital Output Table and their formats.

Device	Supported Address Format	Range
RSX	DO [n] [param]	[n] = 1 to 6
VPR100	DO [n] [param]	[n] = 1 to 24
VRX100	DO [n] [param]	[n] = 1 to 24

Device	Supported Address Format	Range
VRX180	DO [n] [param]	[n] = 1 to 36
UDC5300	DO [n] [param]	[n] = 1 to 4
DR4300	DO [n] [param]	[n] = 1 to 2
DR4500	DO [n] [param]	[n] = 1 to 6
UDC2300	DO [n] [param]	[n] = 1 to 3
UDC3300	DO [n] [param]	[n] = 1 to 3
UMC800	DO [n] [param]	[n] = 1 to 256
HC900	DO [n] [param]	[n] = 1 to 256
TrendView	DO [n] [param]	[n] = 1 to 32

The following table lists the details of the Digital Output Table parameters.

Param	Address Line	Param Format	Access	Devices
Digital Output Value	DO [n] VALUE*	Discrete (bits). [Status Point Only]	RW	RSX, VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500
Digital Output Value	DO [n] VALUE*	Discrete (bits). [Status Point Only]	RO	UDC2300, UDC3300, UMC800, HC900**, TV

* The default Parameter if only "DO [n]" is entered.

** Applies to Rack 1 only, allocation is 16 outputs (bits), 12 slots maximum.

Digital output table

The following table lists the devices which support the Digital Output Table and their formats.

Device	Supported Address Format	Range
DPR180	DO [n] [param]	[n] = 1 to 36
DPR250	DO [n] [param]	[n] = 1 to 48

The following table lists the details of the Digital Output Table parameters.

Param	Address Line	Param Format	Access	Devices
Digital Output Value	DO [n] VALUE*	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

* The default Parameter if only "DO [n]" is entered.

Set point programmer

The following table lists the devices which support the set point programmer and their formats.

Device	Supported Address Format	Range
VPR100	SPP [n] [param]	[n] = 1 to 4
VRX100	SPP [n] [param]	[n] = 1 to 1
VRX180	SPP [n] [param]	[n] = 1 to 4
UDC5300	SPP [n] [param]	[n] = 1 to 1
DR4300	SPP [n] [param]	[n] = 1 to 1
DR4500	SPP [n] [param]	[n] = 1 to 2
UDC2300	SPP [n] [param]	[n] = 1 to 1
UDC3300	SPP [n] [param]	[n] = 1 to 1
UMC800	SPP [n] [param]	[n] = 1 to 4
HC900	SPP [n] [param]	[n] = 1 to 4

The following table lists the details of the set point program parameters.

Param	Address Line	Param Format	Access	Devices
Set Point Programmer Output	SPP [n] OUT*	Floating Point	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Set Point Programmer Segment Time Remaining	SPP [n] SEG_TIME_REM	Floating Point	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Start	SPP [n] START	UINT2	WO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Hold	SPP [n] HOLD	UINT2	WO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Advance	SPP [n] ADV	UINT2	WO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Reset	SPP [n] RESET	UINT2	WO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Status - Ready	SPP [n] STATUS_READY	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - Run	SPP [n] STATUS_RUN	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - Hold	SPP [n] STATUS_HOLD	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - End	SPP [n] STATUS_END	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Set Point Programmer Status - Time Units in Seconds	SPP [n] STATUS_ TIME_UNIT_S	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - Time Units in Minutes	SPP [n] STATUS_ TIME_UNIT_M	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - Time Units in Hours	SPP [n] STATUS_ TIME_UNIT_H	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Elapsed Time	SPP [n] EL_TIME	Floating Point	RO	VPR100, VRX100, VRX180, UDC5300, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - Ramp Rate	SPP [n] STATUS_ RAMP_RATE	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC2300, UDC3300
Set Point Programmer Status - Ramp Units	SPP [n] STATUS_ RAMP_UNITS	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Current Segment Number	SPP [n] SEG_NO	Floating Point	RO	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300
Set Point Programmer Current Segment Number	SPP [n] SEG_NO	Floating Point	RW	UMC800, HC900
Set Point Programmer Status - Type of Hold	SPP [n] STATUS_ HOLD_TYPE	Discrete (bits). [Status Point Only]	RO	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Set Point Programmer Status - Current Segment is a ramp	SPP [n] STATUS_RAMP	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Active Time	SPP [n] ACT_TIME	Floating Point	RO	VPR100, VRX100, VRX180, UDC5300
Set Point Programmer Segment Event #1	SPP [n] EV01	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #2	SPP [n] EV02	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #3	SPP [n] EV03	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #4	SPP [n] EV04	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #5	SPP [n] EV05	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #6	SPP [n] EV06	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #7	SPP [n] EV07	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #8	SPP [n] EV08	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Set Point Programmer Segment Event #9	SPP [n] EV09	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #10	SPP [n] EV10	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #11	SPP [n] EV11	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #12	SPP [n] EV12	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #13	SPP [n] EV13	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #14	SPP [n] EV14	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #15	SPP [n] EV15	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900
Set Point Programmer Segment Event #16	SPP [n] EV16	Discrete (bits). [Status Point Only]	RO	VPR100, VRX100, VRX180, UDC5300, UMC800, HC900

* The default Parameter if only "SPP [n]" is entered.

Set point program additional values

The following table lists the devices which support the Set Point Program Additional Values and their formats.

Device	Supported Address Format	Range
UMC800, HC900	SPP_ADD [n] [param]	[n] = 1 to 4

The following table lists the details of the Set Point Program Additional Values parameters.

Param	Address Line	Param Format	Access	Devices
Current Program Number	SPP_ADD [n] PROG_NO*	Floating Point	RW	UMC800, HC900
Program Save Request	SPP_ADD [n] PROG_SAVE	Floating Point	RW	UMC800, HC900
Auxiliary Output	SPP_ADD [n] AUX_OUT	Floating Point	RO	UMC800, HC900
Guaranteed Soak Low	SPP_ADD [n] SOAK_LOW	Floating Point	RW	UMC800, HC900
Guaranteed Soak High	SPP_ADD [n] SOAK_HIGH	Floating Point	RW	UMC800, HC900
Restart Ramp Rate	SPP_ADD [n] RESTART_RAMP	Floating Point	RW	UMC800, HC900
Display High Range Limit	SPP_ADD [n] DISPLAY_HIGH	Floating Point	RW	UMC800, HC900
Display Low Range Limit	SPP_ADD [n] DISPLAY_LOW	Floating Point	RW	UMC800, HC900
Jog Segment	SPP_ADD [n] JOG_SEG	Floating Point	RW	UMC800, HC900
Loop Start	SPP_ADD [n] LOOP_START	Floating Point	RW	UMC800, HC900
Loop End	SPP_ADD [n] LOOP_END	Floating Point	RW	UMC800, HC900
Repeats	SPP_ADD [n] REPEATS	Floating Point	RW	UMC800, HC900
Time Units	SPP_ADD [n] UNITS_TIME	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Ramp Units	SPP_ADD [n] UNITS_RAMP	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Guaranteed Soak Type	SPP_ADD [n] SOAK_TYPE	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

* The default Parameter if only "SPP_ADD [n]" is entered.

Set point programmer #1 profile segment

The following table lists the devices which support the Set Point Program #1 Profile Segment and their formats.

Device	Supported Address Format	Range
VPR100	SPP1_SEG [n] [param]	[n] = 1 to 63
VRX100	SPP1_SEG [n] [param]	[n] = 1 to 63
VRX180	SPP1_SEG [n] [param]	[n] = 1 to 63
UDC5300	SPP1_SEG [n] [param]	[n] = 1 to 63
DR4300	SPP1_SEG [n] [param]	[n] = 1 to 24
DR4500	SPP1_SEG [n] [param]	[n] = 1 to 12
UDC2300	SPP1_SEG [n] [param]	[n] = 1 to 12
UDC3300	SPP1_SEG [n] [param]	[n] = 1 to 12
UMC800	SPP1_SEG [n] [param]	[n] = 1 to 50
HC900	SPP1_SEG [n] [param]	[n] = 1 to 50

The following table lists the details of the Set Point Program #1 Profile Segment parameters.

Param	Address Line	Param Format	Access	Devices
Ramp/Soak Segment	SPP1_SEG [n] SEG_TYPE 0 = Soak segment 1 = Ramp segment	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #1	SPP1_SEG [n] EV01	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #2	SPP1_SEG [n] EV02	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #3	SPP1_SEG [n] EV03	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #4	SPP1_SEG [n] EV04	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Event #5	SPP1_SEG [n] EV05	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #6	SPP1_SEG [n] EV06	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #7	SPP1_SEG [n] EV07	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #8	SPP1_SEG [n] EV08	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #9	SPP1_SEG [n] EV09	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #10	SPP1_SEG [n] EV10	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #11	SPP1_SEG [n] EV11	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #12	SPP1_SEG [n] EV12	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #13	SPP1_SEG [n] EV13	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #14	SPP1_SEG [n] EV14	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #15	SPP1_SEG [n] EV15	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #16	SPP1_SEG [n] EV16	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Soak Value for Auxiliary Output	SPP1_SEG [n] AUX_OUT	Floating Point	RW	UMC800, HC900
Time	SPP1_SEG [n] TIME*	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate	SPP1_SEG [n] RATE	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Ramp or Soak Value	SPP1_SEG [n] SEG_VALUE	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

* The default Parameter if only "SPP1_SEG [n]" is entered.

Set point programmer #2 profile segment

The following table lists the devices which support the Set Point Program #2 Profile Segment and their formats.

Device	Supported Address Format	Range
VPR100	SPP2_SEG [n] [param]	[n] = 1 to 63
VRX100	SPP2_SEG [n] [param]	[n] = 1 to 63
VRX180	SPP2_SEG [n] [param]	[n] = 1 to 63
UDC5300	SPP2_SEG [n] [param]	[n] = 1 to 63
DR4300	SPP2_SEG [n] [param]	[n] = 1 to 24
DR4500	SPP2_SEG [n] [param]	[n] = 1 to 12
UDC2300	SPP2_SEG [n] [param]	[n] = 1 to 12
UDC3300	SPP2_SEG [n] [param]	[n] = 1 to 12
UMC800, HC900	SPP2_SEG [n] [param]	[n] = 1 to 50

The following table lists the details of the Set Point Program #2 Profile Segment parameters.

Param	Address Line	Param Format	Access	Devices
Ramp/Soak Segment	SPP2_SEG [n] SEG_TYPE 0 = Soak segment 1 = Ramp segment	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #1	SPP2_SEG [n] EV01	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #2	SPP2_SEG [n] EV02	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #3	SPP2_SEG [n] EV03	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #4	SPP2_SEG [n] EV04	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #5	SPP2_SEG [n] EV05	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #6	SPP2_SEG [n] EV06	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #7	SPP2_SEG [n] EV07	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #8	SPP2_SEG [n] EV08	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #9	SPP2_SEG [n] EV09	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #10	SPP2_SEG [n] EV10	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Event #11	SPP2_SEG [n] EV11	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #12	SPP2_SEG [n] EV12	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #13	SPP2_SEG [n] EV13	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #14	SPP2_SEG [n] EV14	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #15	SPP2_SEG [n] EV15	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #16	SPP2_SEG [n] EV16	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Value for Auxiliary Output	SPP2_SEG [n] AUX_OUT	Floating Point	RW	UMC800, HC900
Time	SPP2_SEG [n] TIME*	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate	SPP2_SEG [n] RATE	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Ramp or Soak Value	SPP2_SEG [n] SEG_VALUE	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

* The default Parameter if only "SPP2_SEG [n]" is entered.

Set point programmer #3 profile segment

The following table lists the devices which support the Set Point Programmer #3 Profile Segment and their formats.

Device	Supported Address Format	Range
VPR100	SPP3_SEG [n] [param]	[n] = 1 to 63
VRX100	SPP3_SEG [n] [param]	[n] = 1 to 63
VRX180	SPP3_SEG [n] [param]	[n] = 1 to 63
UDC5300	SPP3_SEG [n] [param]	[n] = 1 to 63
DR4300	SPP3_SEG [n] [param]	[n] = 1 to 24
DR4500	SPP3_SEG [n] [param]	[n] = 1 to 12
UDC2300	SPP3_SEG [n] [param]	[n] = 1 to 12
UDC3300	SPP3_SEG [n] [param]	[n] = 1 to 12
UMC800, HC900	SPP3_SEG [n] [param]	[n] = 1 to 50

The following table lists the details of the Set Point Program #3 Profile Segment parameters.

Param	Address Line	Param Format	Access	Devices
Ramp/Soak Segment	SPP3_SEG [n] SEG_ TYPE 0 = Soak Segment 1 = Ramp Segment	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #1	SPP3_SEG [n] EV01	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #2	SPP3_SEG [n] EV02	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #3	SPP3_SEG [n] EV03	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #4	SPP3_SEG [n] EV04	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #5	SPP3_SEG [n] EV05	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Event #6	SPP3_SEG [n] EV06	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #7	SPP3_SEG [n] EV07	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #8	SPP3_SEG [n] EV08	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #9	SPP3_SEG [n] EV09	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #10	SPP3_SEG [n] EV10	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #11	SPP3_SEG [n] EV11	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #12	SPP3_SEG [n] EV12	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #13	SPP3_SEG [n] EV13	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #14	SPP3_SEG [n] EV14	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #15	SPP3_SEG [n] EV15	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #16	SPP3_SEG [n] EV16	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Value for Auxiliary Output	SPP3_SEG [n] AUX_OUT	Floating Point	RW	UMC800, HC900
Time	SPP3_SEG [n] TIME*	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Rate	SPP3_SEG [n] RATE	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Ramp or Soak Value	SPP3_SEG [n] SEG_VALUE	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

* The default Parameter if only "SPP3_SEG [n]" is entered.

Set point programmer #4 profile segment

The following table lists the devices which support the Set Point Programmer #4 Profile Segment and their formats.

Device	Supported Address Format	Range
VPR100	SPP4_SEG [n] [param]	[n] = 1 to 63
VRX100	SPP4_SEG [n] [param]	[n] = 1 to 63
VRX180	SPP4_SEG [n] [param]	[n] = 1 to 63
UDC5300	SPP4_SEG [n] [param]	[n] = 1 to 63
DR4300	SPP4_SEG [n] [param]	[n] = 1 to 24
DR4500	SPP4_SEG [n] [param]	[n] = 1 to 12
UDC2300	SPP4_SEG [n] [param]	[n] = 1 to 12
UDC3300	SPP4_SEG [n] [param]	[n] = 1 to 12
UMC800, HC900	SPP4_SEG [n] [param]	[n] = 1 to 50

The following table lists the details of the Set Point Program #4 Profile Segment parameters.

Param	Address Line	Param Format	Access	Devices
Ramp/Soak Segment	SPP4_SEG [n] SEG_TYPE 0 = Soak Segment 1 = Ramp Segment	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Event #1	SPP4_SEG [n] EV01	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #2	SPP4_SEG [n] EV02	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #3	SPP4_SEG [n] EV03	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #4	SPP4_SEG [n] EV04	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #5	SPP4_SEG [n] EV05	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #6	SPP4_SEG [n] EV06	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #7	SPP4_SEG [n] EV07	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #8	SPP4_SEG [n] EV08	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #9	SPP4_SEG [n] EV09	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #10	SPP4_SEG [n] EV10	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #11	SPP4_SEG [n] EV11	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #12	SPP4_SEG [n] EV12	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #13	SPP4_SEG [n] EV13	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Event #14	SPP4_SEG [n] EV14	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #15	SPP4_SEG [n] EV15	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #16	SPP4_SEG [n] EV16	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Value for Auxiliary Output	SPP4_SEG [n] AUX_OUT	Floating Point	RW	UMC800, HC900
Time	SPP4_SEG [n] TIME*	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate	SPP4_SEG [n] RATE	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Ramp or Soak Value	SPP4_SEG [n] SEG_VALUE	Floating Point	RW	VPR100, VRX100, VRX180, UDC5300, DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

* The default Parameter if only "SPP4_SEG [n]" is entered.

Set point scheduler values

The following table lists the devices which support the Set Point Scheduler Values and their formats.

Device	Supported Address Format	Range
UMC800	SCHED [n] [param]	[n] = 1 to 1
HC900	SCHED [n] [param]	[n] = 1 to 2

The following table lists the details of the Scheduler Values parameters.

Param	Address Line	Param Format	Access	Devices
Output #1	SCHED [n] OUTPUT1*	Floating Point	RO	UMC800, HC900
Output #1	SCHED [n] OUTPUT2	Floating Point	RO	UMC800, HC900
Output #1	SCHED [n] OUTPUT3	Floating Point	RO	UMC800, HC900
Output #4	SCHED [n] OUTPUT4	Floating Point	RO	UMC800, HC900
Output #5	SCHED [n] OUTPUT5	Floating Point	RO	UMC800, HC900
Output #6	SCHED [n] OUTPUT6	Floating Point	RO	UMC800, HC900
Output #7	SCHED [n] OUTPUT7	Floating Point	RO	UMC800, HC900
Output #8	SCHED [n] OUTPUT8	Floating Point	RO	UMC800, HC900
Auxiliary Output #1	SCHED [n] AUX_ OUT1	Floating Point	RO	UMC800, HC900
Auxiliary Output #2	SCHED [n] AUX_ OUT2	Floating Point	RO	UMC800, HC900
Auxiliary Output #3	SCHED [n] AUX_ OUT3	Floating Point	RO	UMC800, HC900
Auxiliary Output #4	SCHED [n] AUX_ OUT4	Floating Point	RO	UMC800, HC900
Auxiliary Output #5	SCHED [n] AUX_ OUT5	Floating Point	RO	UMC800, HC900
Auxiliary Output #6	SCHED [n] AUX_ OUT6	Floating Point	RO	UMC800, HC900
Auxiliary Output #7	SCHED [n] AUX_ OUT7	Floating Point	RO	UMC800, HC900
Auxiliary Output #8	SCHED [n] AUX_ OUT8	Floating Point	RO	UMC800, HC900
Current Program Number	SCHED [n] PROG_ NO	Floating Point	RW	UMC800, HC900
Current Segment Number	SCHED [n] SEG_NO	Floating Point	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Program Elapsed Time	SCHED [n] EL_ TIME	Floating Point	RO	UMC800, HC900
Segment Time Remaining	SCHED [n] TIME_ REMAIN	Floating Point	RO	UMC800, HC900
Schedule Save Request	SCHED [n] SCHED_ SAVE	Floating Point	RW	UMC800, HC900
Guaranteed Soak Limit #1	SCHED [n] SOAK_ LIMIT_1	Floating Point	RW	UMC800, HC900
Guaranteed Soak Limit #2	SCHED [n] SOAK_ LIMIT_2	Floating Point	RW	UMC800, HC900
Guaranteed Soak Limit #3	SCHED [n] SOAK_ LIMIT_3	Floating Point	RW	UMC800, HC900
Guaranteed Soak Limit #4	SCHED [n] SOAK_ LIMIT_4	Floating Point	RW	UMC800, HC900
Guaranteed Soak Limit #5	SCHED [n] SOAK_ LIMIT_5	Floating Point	RW	UMC800, HC900
Guaranteed Soak Limit #6	SCHED [n] SOAK_ LIMIT_6	Floating Point	RW	UMC800, HC900
Guaranteed Soak Limit #7	SCHED [n] SOAK_ LIMIT_7	Floating Point	RW	UMC800, HC900
Guaranteed Soak Limit #8	SCHED [n] SOAK_ LIMIT_8	Floating Point	RW	UMC800, HC900
Jog Segment	SCHED [n] JOG_ SEG	Floating Point	RW	UMC800, HC900
Event #1	SCHED [n] EVENT_ 01	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #2	SCHED [n] EVENT_ 02	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #3	SCHED [n] EVENT_ 03	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #4	SCHED [n] EVENT_ 04	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #5	SCHED [n] EVENT_ 05	Discrete (bits). [Status Point Only]	RO	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Event #6	SCHED [n] EVENT_06	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #7	SCHED [n] EVENT_07	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #8	SCHED [n] EVENT_08	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #9	SCHED [n] EVENT_09	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #10	SCHED [n] EVENT_10	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #11	SCHED [n] EVENT_11	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #12	SCHED [n] EVENT_12	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #13	SCHED [n] EVENT_13	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #14	SCHED [n] EVENT_14	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #15	SCHED [n] EVENT_15	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Event #16	SCHED [n] EVENT_16	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Status	SCHED [n] STATUS	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Start Schedule	SCHED [n] START	UINT2	WO	UMC800, HC900
Hold Schedule	SCHED [n] HOLD	UINT2	WO	UMC800, HC900
Advance Schedule	SCHED [n] ADVANCE	UINT2	WO	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Reset Schedule	SCHED [n] RESET	UINT2	WO	UMC800, HC900
Time Units	SCHED [n] UNITS_ TIME	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

* The default Parameter if only "SCHED [n]" is entered.

Set point scheduler #1 segment

The following table lists the devices which support the Scheduler #1 Segment and their formats.

Device	Supported Address Format	Range
UMC800, HC900	SCHED1_SEG [n] [param]	[n] = 1 to 50

The following table lists the details of the Scheduler #1 Segment parameters.

Param	Address Line	Param Format	Access	Devices
Soak Type #1	SCHED1_SEG [n] GUAR1*	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Type #2	SCHED1_SEG [n] GUAR2	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Type #3	SCHED1_SEG [n] GUAR3	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Type #4	SCHED1_SEG [n] GUAR4	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Type #5	SCHED1_SEG [n] GUAR5	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Type #6	SCHED1_SEG [n] GUAR6	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Soak Type #7	SCHED1_SEG [n] GUAR7	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Soak Type #8	SCHED1_SEG [n] GUAR8	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #1	SCHED1_SEG [n] EVENT_01	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #2	SCHED1_SEG [n] EVENT_02	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #3	SCHED1_SEG [n] EVENT_03	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #4	SCHED1_SEG [n] EVENT_04	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #5	SCHED1_SEG [n] EVENT_05	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #6	SCHED1_SEG [n] EVENT_06	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #7	SCHED1_SEG [n] EVENT_07	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #8	SCHED1_SEG [n] EVENT_08	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #9	SCHED1_SEG [n] EVENT_09	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #10	SCHED1_SEG [n] EVENT_10	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #11	SCHED1_SEG [n] EVENT_11	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #12	SCHED1_SEG [n] EVENT_12	Discrete (bits). [Status Point Only]	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Event #13	SCHED1_SEG [n] EVENT_13	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #14	SCHED1_SEG [n] EVENT_14	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #15	SCHED1_SEG [n] EVENT_15	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Event #16	SCHED1_SEG [n] EVENT_16	Discrete (bits). [Status Point Only]	RW	UMC800, HC900
Time	SCHED1_SEG [n] TIME Parameter Format:	Floating Point	RW	UMC800, HC900
Output #1	SCHED1_SEG [n] OUTPUT1	Floating Point	RW	UMC800, HC900
Output #2	SCHED1_SEG [n] OUTPUT2	Floating Point	RW	UMC800, HC900
Output #3	SCHED1_SEG [n] OUTPUT3	Floating Point	RW	UMC800, HC900
Output #4	SCHED1_SEG [n] OUTPUT4	Floating Point	RW	UMC800, HC900
Output #5	SCHED1_SEG [n] OUTPUT5	Floating Point	RW	UMC800, HC900
Output #6	SCHED1_SEG [n] OUTPUT6	Floating Point	RW	UMC800, HC900
Output #7	SCHED1_SEG [n] OUTPUT7	Floating Point	RW	UMC800, HC900
Output #8	SCHED1_SEG [n] OUTPUT8	Floating Point	RW	UMC800, HC900
Soak Value for Auxiliary Output #1	SCHED1_SEG [n] AUX_SOAK_1	Floating Point	RW	UMC800, HC900
Soak Value for Auxiliary Output #2	SCHED1_SEG [n] AUX_SOAK_2	Floating Point	RW	UMC800, HC900
Soak Value for Auxiliary Output #3	SCHED1_SEG [n] AUX_SOAK_3	Floating Point	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Soak Value for Auxiliary Output #4	SCHED1_SEG [n] AUX_SOAK_4	Floating Point	RW	UMC800, HC900
Soak Value for Auxiliary Output #5	SCHED1_SEG [n] AUX_SOAK_5	Floating Point	RW	UMC800, HC900
Soak Value for Auxiliary Output #6	SCHED1_SEG [n] AUX_SOAK_6	Floating Point	RW	UMC800, HC900
Soak Value for Auxiliary Output #7	SCHED1_SEG [n] AUX_SOAK_7	Floating Point	RW	UMC800, HC900
Soak Value for Auxiliary Output #8	SCHED1_SEG [n] AUX_SOAK_8	Floating Point	RW	UMC800, HC900
Number of Times to Recycle	SCHED1_SEG [n] RECYCLE	Floating Point	RW	UMC800, HC900
Recycle Segment	SCHED1_SEG [n] RECYCLE_SEG	Floating Point	RW	UMC800, HC900

* The default Parameter if only "SCHED1_SEG [n]" is entered.

Set point scheduler #2 segment

The following table lists the devices which support the Scheduler #2 Segment and their formats.

Device	Supported Address Format	Range
HC900	SCHED2_SEG [n] [param]	[n] = 1 to 50

The following table lists the details of the Scheduler #2 Segment parameters.

Param	Address Line	Param Format	Access	Devices
Soak Type #1	SCHED2_SEG [n] GUAR1*	Discrete (bits). [Status Point Only]	RW	HC900
Soak Type #2	SCHED2_SEG [n] GUAR2	Discrete (bits). [Status Point Only]	RW	HC900
Soak Type #3	SCHED2_SEG [n] GUAR3	Discrete (bits). [Status Point Only]	RW	HC900
Soak Type #4	SCHED2_SEG [n] GUAR4	Discrete (bits). [Status Point Only]	RW	HC900
Soak Type #5	SCHED2_SEG [n] GUAR5	Discrete (bits). [Status Point Only]	RW	HC900
Soak Type #6	SCHED2_SEG [n] GUAR6	Discrete (bits). [Status Point Only]	RW	HC900
Soak Type #7	SCHED2_SEG [n] GUAR7	Discrete (bits). [Status Point Only]	RW	HC900
Soak Type #8	SCHED2_SEG [n] GUAR8	Discrete (bits). [Status Point Only]	RW	HC900
Event #1	SCHED2_SEG [n] EVENT_01	Discrete (bits). [Status Point Only]	RW	HC900
Event #2	SCHED2_SEG [n] EVENT_02	Discrete (bits). [Status Point Only]	RW	HC900
Event #3	SCHED2_SEG [n] EVENT_03	Discrete (bits). [Status Point Only]	RW	HC900
Event #4	SCHED2_SEG [n] EVENT_04	Discrete (bits). [Status Point Only]	RW	HC900
Event #5	SCHED2_SEG [n] EVENT_05	Discrete (bits). [Status Point Only]	RW	HC900

Param	Address Line	Param Format	Access	Devices
Event #6	SCHED2_SEG [n] EVENT_06	Discrete (bits). [Status Point Only]	RW	HC900
Event #7	SCHED2_SEG [n] EVENT_07	Discrete (bits). [Status Point Only]	RW	HC900
Event #8	SCHED2_SEG [n] EVENT_08	Discrete (bits). [Status Point Only]	RW	HC900
Event #9	SCHED2_SEG [n] EVENT_09	Discrete (bits). [Status Point Only]	RW	HC900
Event #10	SCHED2_SEG [n] EVENT_10	Discrete (bits). [Status Point Only]	RW	HC900
Event #11	SCHED2_SEG [n] EVENT_11	Discrete (bits). [Status Point Only]	RW	HC900
Event #12	SCHED2_SEG [n] EVENT_12	Discrete (bits). [Status Point Only]	RW	HC900
Event #13	SCHED2_SEG [n] EVENT_13	Discrete (bits). [Status Point Only]	RW	HC900
Event #14	SCHED2_SEG [n] EVENT_14	Discrete (bits). [Status Point Only]	RW	HC900
Event #15	SCHED2_SEG [n] EVENT_15	Discrete (bits). [Status Point Only]	RW	HC900
Event #16	SCHED2_SEG [n] EVENT_16	Discrete (bits). [Status Point Only]	RW	HC900
Time	SCHED2_SEG [n] TIME	Floating Point	RW	HC900
Output #1	SCHED2_SEG [n] OUTPUT1	Floating Point	RW	HC900
Output #2	SCHED2_SEG [n] OUTPUT2	Floating Point	RW	HC900
Output #3	SCHED2_SEG [n] OUTPUT3	Floating Point	RW	HC900

Param	Address Line	Param Format	Access	Devices
Output #4	SCHED2_SEG [n] OUTPUT4	Floating Point	RW	HC900
Output #5	SCHED2_SEG [n] OUTPUT5	Floating Point	RW	HC900
Output #6	SCHED2_SEG [n] OUTPUT6	Floating Point	RW	HC900
Output #7	SCHED2_SEG [n] OUTPUT7	Floating Point	RW	HC900
Output #8	SCHED2_SEG [n] OUTPUT8	Floating Point	RW	HC900
Soak Value for Auxiliary Output #1	SCHED2_SEG [n] AUX_SOAK_1	Floating Point	RW	HC900
Soak Value for Auxiliary Output #2	SCHED2_SEG [n] AUX_SOAK_2	Floating Point	RW	HC900
Soak Value for Auxiliary Output #3	SCHED2_SEG [n] AUX_SOAK_3	Floating Point	RW	HC900
Soak Value for Auxiliary Output #4	SCHED2_SEG [n] AUX_SOAK_4	Floating Point	RW	HC900
Soak Value for Auxiliary Output #5	SCHED2_SEG [n] AUX_SOAK_5	Floating Point	RW	HC900
Soak Value for Auxiliary Output #6	SCHED2_SEG [n] AUX_SOAK_6	Floating Point	RW	HC900
Soak Value for Auxiliary Output #7	SCHED2_SEG [n] AUX_SOAK_7	Floating Point	RW	HC900
Soak Value for Auxiliary Output #8	SCHED2_SEG [n] AUX_SOAK_8	Floating Point	RW	HC900
Number of Times to Recycle	SCHED2_SEG [n] RECYCLE	Floating Point	RW	HC900
Recycle Segment	SCHED2_SEG [n] RECYCLE_SEG	Floating Point	RW	HC900

* The default Parameter if only "SCHED2_SEG [n]" is entered.

Non-numbered addresses

The following table lists the details of the Non-numbered Address parameters.

Param	Address Line	Param Format	Access	Devices
Relay #1	RELAY1	Discrete (bits). [Status Point Only]	RO	DR4300
Relay #2	RELAY2	Discrete (bits). [Status Point Only]	RO	DR4300
Alarm Relay #1	ALMRLY1	Discrete (bits). [Status Point Only]	RO	DR4500
Alarm Relay #2	ALMRLY2	Discrete (bits). [Status Point Only]	RO	DR4500
Control Relay #1	CR1	Discrete (bits). [Status Point Only]	RO	DR4500
Control Relay #2	CR2	Discrete (bits). [Status Point Only]	RO	DR4500
Control Relay #3	CR3	Discrete (bits). [Status Point Only]	RO	DR4500
Control Relay #4	CR4	Discrete (bits). [Status Point Only]	RO	DR4500
Control Relay	CR	Discrete (bits). [Status Point Only]	RO	UDC2300, UDC3300
Alarm Relay #2	ALMRLY2	Discrete (bits). [Status Point Only]	RO	UDC2300, UDC3300
Alarm Relay #1	ALMRLY1	Discrete (bits). [Status Point Only]	RO	UDC2300, UDC3300
INSTMODE	INSTMODE	Floating Point	RW	RSX, VPR100, VRX100, VRX180, UDC5300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
CONFIG_CLEAR	CONFIG_CLEAR	Floating Point	WO	RSX, VPR100, VRX100, VRX180, UDC5300, HC900
LOAD_RECIPE	LOAD_RECIPE	Floating Point	WO	UMC800
CHART_SPEED	CHART_SPEED	Floating Point	RO	DR4300, DR4500
Pen #1 High Value	PEN1HI	Floating Point	RO	DR4300, DR4500
Pen #1 Low Value	PEN1LO	Floating Point	RO	DR4300, DR4500
Number of Chart Divisions	CHART_DIVS	Floating Point	RO	DR4500
CHART_STATUS	CHART_STATUS	Floating Point	RO	DR4500
Pen #1 Status	PEN1STAT	Floating Point	RO	DR4500
Pen #2 Status	PEN2STAT	Floating Point	RO	DR4500
Pen #2 High Value	PEN2HI	Floating Point	RO	DR4500
Pen #2 Low Value	PEN2LO	Floating Point	RO	DR4500
Pen #3 Status	PEN3STAT	Floating Point	RO	DR4500
Pen #3 High Value	PEN3HI	Floating Point	RO	DR4500
Pen #3 Low Value	PEN3LO	Floating Point	RO	DR4500
Pen #4 Status	PEN4STAT	Floating Point	RO	DR4500
Pen #4 High Value	PEN4_HIGH	Floating Point	RO	DR4500
Pen #4 Low Value	PEN4_LOW	Floating Point	RO	DR4500

HC900 and UMC800 SPP and recipe support

6

Overview

The HC900 and UMC800 SPP & Recipe Support is an application that enables you to configure and control Set Point (SP) programmers and variables in one or more HC900 and/or UMC800 controllers through Station. The application allows operators to easily configure set point profiles and Variable-based recipes offline, before downloading to a specific controller. Also supported is the monitoring and configuration of running set point programs. The HC900/UMC800 application provides an easy alternative to configuring, monitoring, and loading SP programs and recipes from the controller operator interface.

In particular, the HC900/UMC800 SPP and Recipe Support includes:

- Configuration and maintenance of recipe definitions using Variables in Station.
- Downloading recipes to HC900 and UMC800 controllers.
- Configuration and maintenance of SP profiles through Station displays.
- Configuration and maintenance of combined recipe definitions in Station. A combined recipe includes a recipe with a defined list of Variables and/or up to two SP profiles.
- Download a combined recipe to a compatible HC900 or UMC800 controller. (In an HC900 controller, profiles may only be sent to the first four programmers.)
- Upload and download of SP profiles between the server database and HC900/UMC800 SP programmers. (In an HC900 controller, profiles can be sent to the first four programmers only.)
- View and modify online the first four HC900/UMC800 SP programmers in a controller (configuration and “current state”).

Planning

This section describes the planning and design-related issues concerned with configuring HC900 and UMC800 SPP and Recipe Support. After reading this section, you will be able to plan for the configuration process.

Resource requirements

This section details the requirements and restrictions for the HC900/UMC800 application.

Set point profile and recipe slots

The server database allows you to configure and store up to 1000 SP profiles. These profiles can be downloaded to HC900 and UMC800 SP programmers in the same manner as profiles stored locally in the controller.

The system overwrites Profiles 1 to 4 in the HC900's and the UMC800's own pool of stored profiles. Apart from these four profiles, it is possible, although strongly not recommended, to use the remaining profile slots internal to the controller in parallel with the 1000 server database profiles.

The server database also allows you to configure and store up to 1000 recipes. These recipes can then be downloaded to HC900 and UMC800 controllers in the same manner as recipes stored locally in the controller.

The system overwrites Recipe 1 in the HC900's and the UMC800's own pool of stored recipes. Apart from this recipe, it is possible, although strongly not recommended, to use the remaining recipe slots in parallel with the 1000 server database recipes.

Set point program history

The history of a SP program can be viewed on a standard trend and compared to its ideal pre-plotted profile. To collect history, a point needs to be built for each programmer in an HC900 and a UMC800 controller. These points are used to monitor the primary and auxiliary PV outputs of the processes driven by the programmers, collecting the values and storing them in history.

Note that only the first four programmers in an HC900 can be monitored. This means that a maximum of four points, one for each programmer, are required for each HC900 and UMC800 controller in the system.

Display locking

For safety reasons and data integrity, recipes and SP programmers can only be configured and maintained by one user at a time. Any users who try to access these displays while they are in use are locked out. A message indicating the lockout is displayed, indicating the Station number that is currently using the display. These displays remain locked until the Station either exits the displays or is disconnected.

Migration/conversion requirements

The 1000 SP profiles and recipes stored in the server database supersede the HC900's and the UMC800's own set of stored profiles and recipes. This section details how to migrate the existing profiles and recipes into the server database from a controller.

Set point profiles

To migrate existing profiles from a controller, a utility transfers the stored profiles to a block within the server database of 1000 profiles.

The utility is named `umc800export`, and may be run from the command line:

To migrate profiles:

- 1 From a command line enter
C:> umc800export
 The UMC800 Profile Export Utility starts

- 2 Enter a valid controller number.
- 3 Enter the starting profile number.
- 4 Enter y to proceed.

For example:

```
**** UMC800 Profile Export Utility ****
Enter valid controller number: 1
Enter profile number to start from (1 to 931): 1
All profiles in Controller 1 will exported to profiles 1 to 70
Do you want to proceed (Y/N) ? y
Profile 70 of 70
Exported all profiles
C:>
```

Recipes

No utility exists to transfer existing recipes from an HC900 or a UMC800 controller to the server database of 1000 recipes. Recipes need to be re-created manually in Station.

Configuration

In this section, you will learn how to configure HC900 and UMC800 recipes, SP profiles, and combined recipes. Configuration requirements for setting up the set point programmer monitoring displays are also presented.

Prerequisites

Before configuring the HC900/UMC800 SPP and Recipe Support, ensure that you have:

- Access to the MNGR operator account in Station.
- Fast and Extended history.
- SPP function blocks configured in each HC900 or UMC800 controller. You will need one block for each SP programmer (see *UMC800 Control Builder User's Guide* or HC900's *HC Designer User's Guide* or Help files for information on function blocks).

Considerations

Each recipe, SP profile, and combined recipe stored in the server database must have a unique name (respectively).

SP profiles must have zero length/rate segments only at the end of the profile.

Configuring a recipe

A recipe is a collection of 50 Variable signal tags and their values or states. Each Variable is either a digital or analog element in a control configuration, acting as an input to any connected function blocks. When a recipe is loaded, the values or states of the signal tags in the recipe replace the values of those signals in the controller's configuration.

Up to 1000 recipes can be created and maintained using the Station HC900 and UMC800 Recipe Configuration displays.

To configure a recipe:

- 1 In Station select **Configure > Applications > HC900/UMC800 > Recipes (Variables Only)**. The Recipe Selection display opens.
- 2 Click the recipe that you want to configure or modify, or click a blank slot to create a new recipe.
- 3 Click on the recipe name to load its configuration.

When the Recipe Configuration display opens, the server attempts to read a list of all variables from the currently selected “Compatible” controller. If the controller is not a valid HC900 or UMC800 controller or the upload fails, an alarm is raised.

The variable list does not overwrite any of the variables configured in the current recipe, nor do variables in the recipe need to be members of the list. Instead, the list is used to provide default selections in the “Variable” boxes to help when configuring a recipe.

By changing the controller selection from the “Compatible controller” dropdown list, the server attempts to read a new list of all variables from the controller. If the controller is not a valid HC900 or UMC800 controller or the upload fails, an alarm is raised.



Note

Only the first 188 variables configured on an HC900 are used to populate the variable combobox. While additional variables are not visible in the combobox, they can still be added to a recipe.

Download to controller

Allows the user to download the current recipe to an HC900 or a UMC800 controller. A recipe can be downloaded to any controller, not just the “Compatible controller”. See the section “Downloading a recipe” on page 113 for information on downloading a recipe.

See the section “Configuring a combined recipe” on page 108 for information on configuring a recipe for use in a combined recipe.

Configuring a SP profile

A SP profile is a time-based program typically used as the set point of a control loop. Each program may be from 2 to 50 segments in length, where each segment of the program may be a ramp or soak except the last segment that must be a soak.

In addition to the main output value, a second analog value is available for each step of the program. This output is a fixed soak value, which may be used as an input to another function or to provide a set point for a secondary control loop in the process such as pressure or % carbon.

A set point guarantee function is provided that holds the program if a process variable exceeds a predefined deviation from the set point. The set point guarantee can be selected to be active for the entire program, for soaks only, or for user specified segments.

Up to 1000 profiles can be created and maintained using the Station HC900 and UMC800 Profile Configuration displays.

To configure a SP profile, perform the following steps:

- 1 In Station select **Configure > Applications > HC900/UMC800 > Set Point Programs > Profile Setup**. The Profile Selection display opens.
- 2 Select the profile that you want to configure or modify, or click a blank slot to create a new profile.
- 3 Click the profile name to load its configuration.

The Profile Configuration display allows all the details of a SP profile to be edited from a single display. Changes made to configuration are applied immediately to the stored profile, but do not have any effect on profiles that are currently loaded into HC900 and UMC800 controllers.

Program control

Values such as Restart Rate and Loop Segment control the dynamic execution of a program. These values can be shown or hidden using the Show/Hide button.

Clone a profile

Allows the user to copy all the details of another of the 1000 stored profiles to the current profile slot. The “Name” field is not copied and made blank.

Upload from controller

Allows the user to upload the profile currently loaded in an HC900 or UMC800 SP programmer into the profile slot currently being edited.

Download to controller

Allows the user to download the current profile to an HC900 or UMC800 SP programmer. Note that this action causes the selected programmer to be cleared and reset before the profile is downloaded. Any pre-existing program is aborted and overwritten. See the section “Downloading a SP profile” on page 114 for information on downloading a profile.

Configuring a combined recipe

A combined recipe is a combination of a recipe, up to two set point profiles and a list of “Compatible Destinations”.

Each combined recipe can be associated with a number of destinations, any one of which can be selected by the operator as a target for the combined recipe. Each destination includes an HC900 or UMC800 controller, a set point programmer for each profile in the combined recipe, and an optional ‘variable suffix’. This suffix is appended to every variable in the recipe component of a combined recipe, before it is sent to a controller. This allows the same recipe to be used for more than one set of variables in a single HC900 or UMC800 controller if the controller

is used to control multiple, similar processes. It is up to the user to configure the Variable tag names with the proper suffixes in the controller configuration so that the recipe with values for the Variables with these suffixes can be loaded from the server database. An error is posted if these Variable tag names are not found on download.

When a combined recipe is loaded to a controller, the SPP profiles are loaded into the specified programmers and the recipe is loaded to the controller's configuration.

Up to 1000 combined recipes can be created and maintained using the Station HC900/UMC800 Combined Recipe Configuration displays.

To configure a combined recipe, perform the following steps:

- 1 In Station select **Configure > Applications > HC900/UMC800 > Combined Recipes**. The Combined Recipe Selection display opens.
- 2 Select the combined recipe that you want to configure or modify, or click a blank slot to create a new combined recipe.
- 3 Click on its name to load the combined recipe.

The Combined Recipe Configuration display allows combined recipes to be configured and stored in the server database. Changes made to configuration are applied immediately to the stored combined recipe, but do not have any immediate effect on profiles or variable values currently loaded in HC900 or UMC800 controllers.

There are three optional components to a combined recipe. The first is a recipe selected from the 1000 recipes stored in the server database (see the section “Configuring a recipe” on page 106 for information on recipes). The remaining components are up to two SP profiles, selected from the 1000 profiles stored in the server database (see the section “Configuring a SP profile” on page 107 for information on SP profiles). A combined recipe may include any, some, or none, of these components.

Destination list

Each combined recipe may be configured with up to twenty “Compatible Destinations”. This allows a single combined recipe to drive a number of processes in a given plant. For example, the same combined recipe may be used to operate three furnaces – where a different SP programmer in a controller, and a different set of variables, control each furnace. The recipe Variable suffix allows the same Combined Recipe to be directed to another set of Variables with the same function for a similar process in the controller.

Name

Each destination may be given a name to more easily identify the process it drives.

Controller

Each destination has a controller to which each component of the combined recipe is downloaded.

Prog A & B

These identify the SP programmers in the destination controller to which profiles A and B will be downloaded.

Var. Suffix

Identifies a short string that will be appended to every variable name in the recipe component of a combined recipe before it is downloaded. This allows the same recipe to be loaded to a number of subsets of variables within the same controller.

For example, assume the recipe contains the variables TEMP, VOLUME and PRESS. If destination “FURNACE1” has a variable suffix of “1” and destination “FURNACE2” has a variable suffix of “2”, then when the combined recipe is downloaded to “FURNACE2”, the variables updated will be TEMP2, VOLUME2 and PRESS2. If the destination had been “FURNACE1”, then TEMP1, VOLUME1 and PRESS1 would have been updated.

Download

See the section “Downloading a combined recipe” on page 114 for information on downloading a combined recipe.

Configuring SPP monitoring

The user may view and control the current state of set point programs in the HC900 and UMC800 controllers from one of three monitoring displays. The SPP Summary displays allows the user to monitor the first four programmers in a given HC900 or UMC800 controller. This display provides information about the SP programmers, including their current state and segment number, the segment time remaining, and a history of the current program.

The SPP Program display allows the user to view the program configuration of a specific programmer. This display is very similar to the Profile Configuration display in that it shows a time-based program of 2 to 50 segments in length, where each segment of the program can be a ramp or soak except the last segment that must be a soak. The difference is that the SPP Program display reads and writes a set point program from a SP programmer, and does not store the program in the server database.

The SPP Trend display allows the user to view the history of a SP programmer and compare it to the ideal profile. To collect history, a point needs to be built for each SP programmer in a controller. These points are used to monitor the process PVs driven by the primary and auxiliary outputs of the programmers, collecting the values and storing them in history.

Building points for SPP monitoring

Quick Builder can be used to build the points for monitoring the SP programmers. The points must be of “Analog” type, and a unique point must be created for each programmer. The source addresses used to monitor SP Programmer 1 in an HC900 or UMC800 controller are described below.

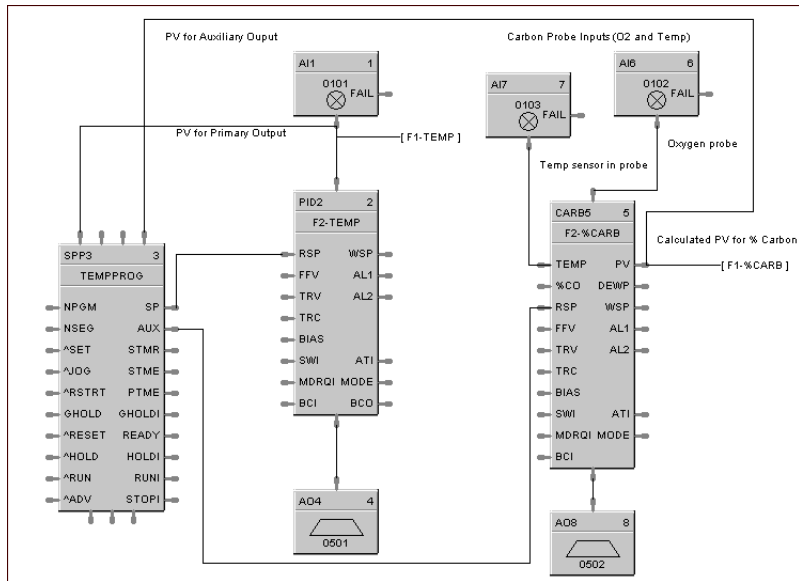
Table 1 SP Programmer 1 Parameter Definition

Parameter	Source Address
PV	Address the PV being driven by the output of SPP 1 in your process. See below for an example.
AL1	PV high value
AL2	PV low value
SP	SPP 1 OUT
A1	Address the PV being driven by the auxiliary output of SPP 1 in your process. See below for an example.
AL3	A1 high value
AL4	A1 low value
A2	SPP_ADD 1 AUX_OUT
A3	SPP 1 STATUS_HOLD
A4	SPP 1 STATUS_END

The point should also be configured with:

- Two second scan periods for each parameter.
- Disable Alarming set (that is, alarms are disabled).
- Fast, Normal, and Extended history collection for **each** parameter.
- PV range sufficient to cover the output of the programmer.

The following diagram illustrates a typical HC900/UMC800 configuration. In this example, when configuring a point in Station to track programmer block SPP3, you should configure the point’s PV parameter to read the PV of loop PID2, and it’s A1 parameter to read the calculated PV from CARB5.

Figure 11 Example SPP Implementation

To monitor the other SP programmers, create a new point for each programmer and replace the '1' in the Source Address with the given programmer number (valid 1 to 4). Each point must have a unique name. Repeat this process until you have created points for each programmer. When all points have been built, download them to the server database. See the *Configuration Guide* for information on points.

To configure SPP monitoring:

- 1 Disable the HC900 & UMC800 channels.
- 2 Select **Configure > Applications > HC900/UMC800 > Programmer Operation**. The SPP Summary display opens.
- 3 For each HC900 and UMC800 controller, enter each point configured for this controller in the appropriate slot.
- 4 Enable the channels.

You can verify the SPP monitoring by checking that the primary and auxiliary SP follow that of the programmers (displayed on the controller faceplate).

Operation

This section describes how to use the HC900/UMC800 SPP & Recipe Support on a routine basis. Standard tasks include downloading recipes and SP profiles, and issuing commands to the SP programmers. After reading this section, you will be able to control HC900 and UMC800 controllers from Station.

Prerequisites

It is assumed that you have successfully completed the configuration procedure detailed in the previous section and that all prerequisites have been met.

Procedure

The HC900/UMC800 application allows you to easily perform routine control tasks from Station, including:

- Downloading recipes
- Downloading SP profiles
- Downloading combined recipes
- Controlling a SP programmer

Downloading a recipe



Caution

When you download a recipe, you are in effect writing new values to the variables. Be aware that by changing the variable values, you can affect running programs if they use the variables as inputs.

To download a recipe:

- 1 In Station select **Configure > Applications > HC900/UMC800 > Recipes (Variables Only)**. The Recipe Selection display opens.
- 2 Click the recipe that you want to configure or modify, or click a blank slot to create a new recipe.
- 3 Click on the recipe name to load its configuration.
- 4 Click the **Download to Controller** button and select a controller destination. Note that a recipe can be downloaded to any controller, not just the “Compatible controller”.
- 5 Click **OK** to accept the current controller selection. A confirmation dialog box appears.

- 6 Click **Download** to accept the recipe destination or Cancel to remove the dialog box. The message “Downloading recipe...” appears.
If successful, the message “Recipe download complete.” appears. Otherwise “Recipe download failed.” is displayed and an alarm is raised. See the section “Troubleshooting” on page 117 for possible fail reasons.

Downloading a SP profile



Caution

Downloading a profile will cause the selected programmer to be cleared and reset before the profile is downloaded. Any pre-existing program will be aborted and overwritten.

To download a SP profile:

- 1 In Station select **Configure > Applications > HC900/UMC800 > Set Point Programs > Profile Setup**. The Profile Selection display opens.
- 2 Click the profile that you want to configure or modify, or click a blank slot to create a new profile.
- 3 Click on the profile name to load its configuration.
- 4 Click the **Download to Controller** button and select a controller and programmer destination from the dialog box.
- 5 Click **OK** to accept the current controller and programmer selection. A confirmation dialog box appears.
- 6 Click **Download** to accept the profile destination or Cancel to remove the dialog box. The message “Downloading profile...” appears.
If successful, the message “Profile download complete.” appears. Otherwise “Profile download failed.” appears and an alarm is raised. See the section “Troubleshooting” on page 117 for possible fail reasons.

Downloading a combined recipe



Caution

If the download includes a recipe, then running programs can be affected by changing the variable values. If the download includes a profile, then the selected programmer(s) will be cleared and reset before the profile is downloaded. Any pre-existing program(s) will be aborted and overwritten.

To download a combined recipe:

- 1 In Station select **Configure > Applications > HC900/UMC800 > Combined Recipes**. The Combined Recipe Selection display opens.
- 2 Click the combined recipe that you want to configure or modify, or click a blank slot to create a new combined recipe.
- 3 Click on the combined recipe name to load its configuration.
- 4 Click the **Download** button to download the combined recipe. Select a controller destination and click on its “Download” button. A confirmation dialog box appears.
- 5 Click **OK** to accept the combined recipe destination or Cancel to remove the dialog box. The message “Downloading combined recipe...” appears.
If successful, the message “Combined recipe download complete.” appears. Otherwise “Combined recipe download failed.” appears and an alarm is raised. See the section “Troubleshooting” on page 117 for possible fail reasons.

Controlling a SP programmer

Considerations

- You can only change the present segment while the program is in operation. You must first place the programmer in the Hold state, then you may alter the Start Value/Soak Value or Rate/Soak Time. You must then select the Start action for the program to continue.
- Not all commands are valid in all programmer states. For example, ‘Clear’ is not valid when the programmer is in ‘Run’. You must also go to the Hold state first before Advance or Reset. You can only start the program in the Ready state. Commands can also be issued from the SPP Trend page.

To control a SP programmer:

- 1 In Station select **Configure > Applications > HC900/UMC800 > Programmer Operation**. The SPP Summary display opens.
- 2 Select a controller in the combobox. The display updates with the current state of the SP programmers configured.
- 3 Select the programmer that you want to control. Click on programmer’s number to load the SPP Program page with its configuration. This is as shown below.
- 4 Click **Command Programmer**. The Select Action dialog box appears. Select the required action and click **OK**.

- 5 A confirmation dialog box appears. Click Ok to accept the action or Cancel to remove the dialog box.

If the command is successful, the message “Command sent.” appears and the SP programmer status changes to reflect the command. Otherwise “Failed to send command.” appears. See the section “Troubleshooting” on page 117 for possible fail reasons.

While the program is running, the present segment number is highlighted and the segment and elapsed timers are active. When in Hold, the segment timer stops but the elapsed timer continues.

Click the Trend icon at the top right of the SPP Program display to access the SPP Trend display. If a profile has been downloaded to the programmer, a SP pre-plot for the Primary programmer output appears. The time of the program is spread over a single screen for this plot in hours or minutes, depending on the time units. Alternatively you can also select the Auxiliary output plot (if configured).

You can operate the programmer using the Command Programmer button as described for the SPP Program display. The status information includes the event LEDs that are red when the event is ON.

When the program is in Hold, the PV plotting stops. The PV continues plotting when the program is re-started.

Troubleshooting

This section describes cross-checks and remedies to perform if HC900/UMC800 SPP & Recipe Support does not respond as anticipated.

Behavior	Things to try or confirm
Cannot use Station to control an HC900 or UMC800. The commands appear to have no effect.	Ensure that the application has been installed correctly and that all prerequisites have been met. Make sure the UMC800SP.EXE task is running.
Downloading/uploading a stored recipe or SP profile fails and causes an alarm to be raised in Station.	Ensure the selected controller is a valid HC900/UMC800. Check that the server can communicate with the controller, that is, the controller status is OK. If performing a download, ensure the target programmer has a SPP function block. Ensure the controller is in 'Run' mode (set on the controller hardware). Check the server log for error messages.
The "Clone a Profile" dialog box does not let me select the correct profile.	Check that each profile has a unique name. If this is not the case, then the dialog box will only select the first profile and clone this one.
Cannot enter a point name on the SPP Summary page.	Ensure that the HC900 and UMC800 channel(s) are out of service when entering the point names.
Downloading a program from the SPP Program page fails and causes an alarm to be raised in Station.	As per "Download profile" (above). The SP programmer must be in 'Ready' state to edit segments. Ensure all parameter have valid values for example, restart rate and jog segment not zero. Check the server log for error messages.
The command issued to a SPP programmer appears to have no effect.	Some actions require the SP programmer to be in a certain state, for example, 'Clear' is not valid when the programmer is in 'Run'.

Behavior	Things to try or confirm
The trend does not display the program history or the ideal profile.	Check that a point has been built and specified for the SP programmer. Ensure that HC900 and UMC800 channel(s) are enabled and the point has “Scanning and Control enabled” set. Ensure you are licensed for Fast and Extended history collection. Check that the point parameters have been configured properly and are collecting history. Make sure the point is not in alarm. The point range should be large enough to cover the output of the programmer.
The program history does not look like the ideal profile.	An ‘Advance’ command causes the programmer to advance to the next segment. This causes a ‘gap’ in the history values and results in the running program to be ‘distorted’.
The trend draws fewer segments than in the SP program.	The end of the program is taken as the first segment with a length/rate of zero. Ensure that your program only contains these types of segments at the end of the program. To check this, you can upload the program in the SPP Program page. Check the server log for error messages.

Index

A

Address
 Named 28
 Non-named 33
architecture 10

B

Black Box Connections 11

C

Channel
 Write Delay 20
channel definition 18
combined recipes, configuring 108
configuration wizard 8
configuring Universal Modbus, steps for 5
connecting controllers to Universal Modbus 10
controller
 definition 24

D

data formats 33
documentation for Universal Modbus 6

F

formats, data 33

H

HC900 recipes 106

L

location syntax 28

P

Port tab 20

R

recipes
 configuring 106
 downloading 113
RS-232 10
RS-485 12

S

scanning
 optimizing for Universal Modbus 36
set point profiles
 configuring 107
 downloading 114
 migrating 104
set point programs 110
SPP monitoring 110
Stallion 12

U

UMC800 recipes 106

W

wizard, configuration 8

Modbus Interface Reference

Issue

3

Date

November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for Modbus	6
	Device support.	6
	Other documentation on Modbus	6
2	Modbus setup	7
	Architectures for Modbus	8
	Serial connection	8
	Modbus Plus connection	12
	Modbus TCP connection	14
	Modbus function codes	15
	Modbus data tables	15
	Setting up a Modbus controller	17
	Setting up a Modbus Plus network	17
	Serial communication settings.	18
3	Modbus configuration and addressing	19
	Defining a Modbus channel	20
	Modbus channel main properties.	20
	Modbus channel port properties	22
	Modbus channel redundant port properties.	25
	Defining a Modbus controller	26
	Modbus controller main properties	26
	Defining a Modbus address for a point parameter	31
	Entering an address	31
	Location syntax	31
	Optimizing scanning performance	35
	Reducing the number of scan packets	35
	Reducing the number of data tables being scanned.	36
4	Server tasks for Modbus	37
	Testing Modbus to field device communications	38
	Troubleshooting point configuration errors.	39

Getting started

1

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



Note

This reference was previously called the *Modicon Controller Reference*.

Steps for connecting and configuring a Modbus.

Complete each step before commencing the next step.

Step:	Go to:
Setup the controller and network	page 17
Define channels using Quick Builder	page 20
Define controllers using Quick Builder	page 26
Download channel and controller definitions to the server	
Test communications	page 38
Define controller points using Quick Builder	page 31

Support and documentation for Modbus

Device support

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 IBM Host Based Devices User's Guide*
- *Modbus Plus Host Interface Device Driver Manual for Windows NT*

Modbus setup

2

This chapter describes how to set up a Modbus controller.

The tasks for setting up a Modbus controller are:

For:	Go to:
Supported architectures	page 8
Setting up a controller	page 17

Architectures for Modbus

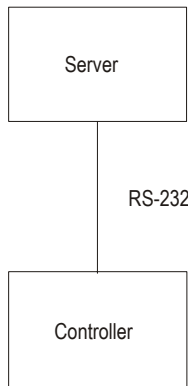
The server supports both single and redundant communications to Modbus controllers via serial connection, a *Modbus Plus* connection (Modbus Plus is Modbus's Local Area Network), or a Modbus TCP connection.

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

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

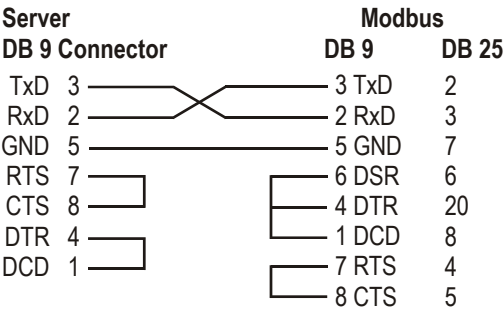
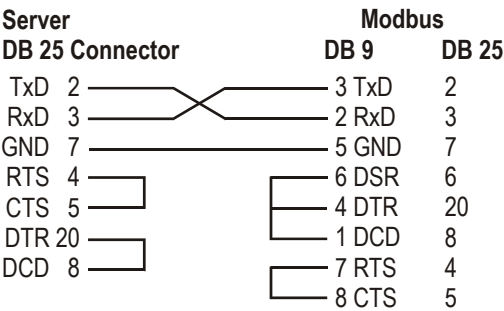


Figure 3 RS-232 wiring details for a server with a DB 25 connector

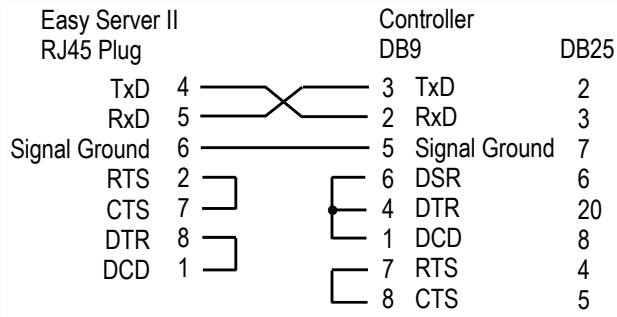


Connections using interface cards

The following figures show the wiring details when using Stallion EasyServer or Emulex cards.

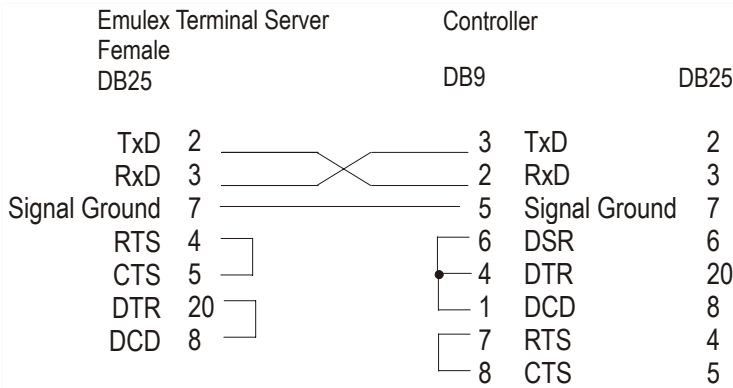
Stallion EasyServer II connection

Figure 4 No handshaking cable



Emulex connection

Figure 5 No handshaking cable



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 6 RS-422 multi-drop connection

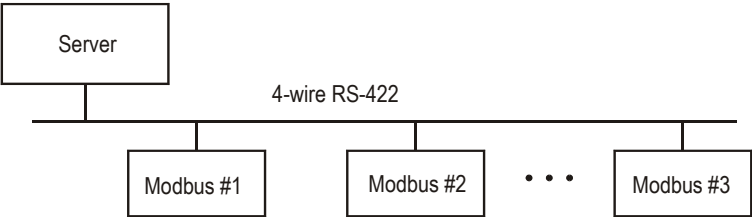
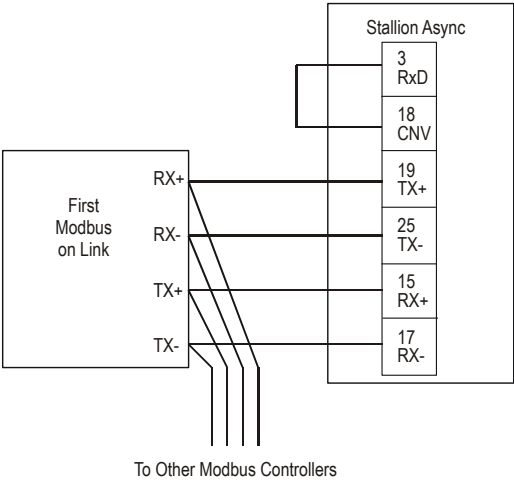


Figure 7 RS-422 wiring details



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 8 RS-485 multi-drop connection

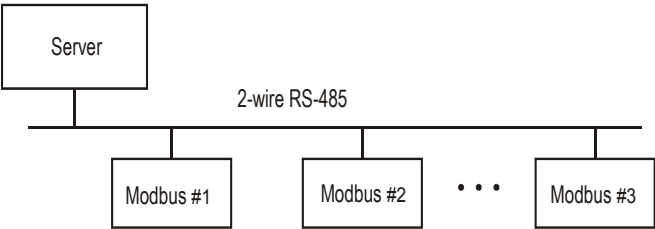
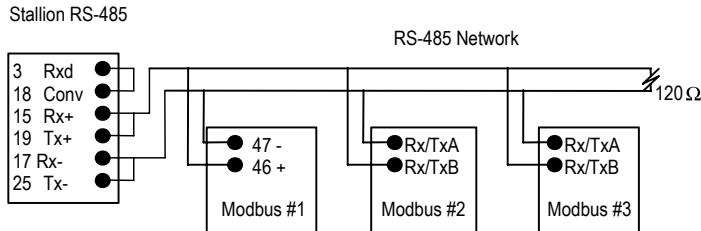
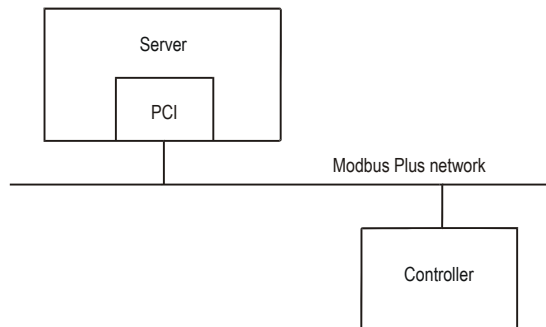


Figure 9 RS-485 wiring details

Modbus Plus connection

To connect Modbus controllers to the server via a Modbus Plus network, you require the following hardware and software:

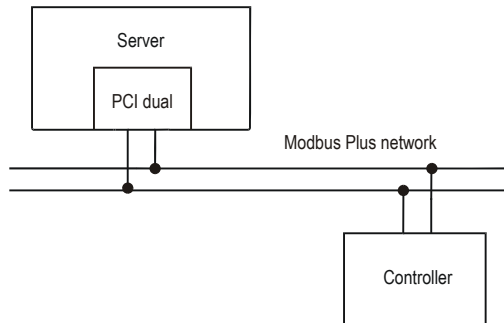
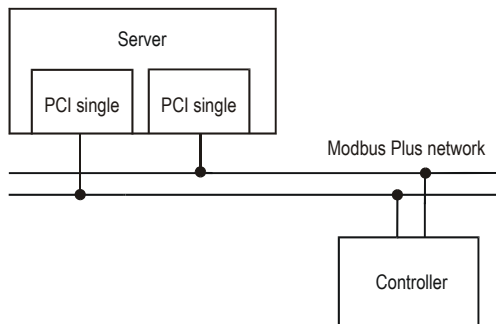
- Modbus Plus PCI Network Interface Adapter, which is installed in the server (Schneider Electric part number 416NHM30030A or SA85000, for single links, 416NHM30032A or SA85002 for redundant links).
- Modbus Plus Driver supplied with the PCI card.

Figure 10 Non-redundant Modbus Plus network architecture

Redundant communication architectures

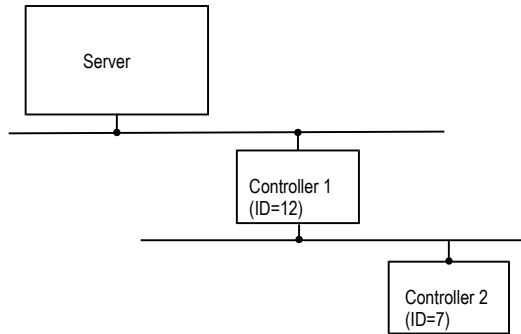
If you require redundant communications, there are two options:

- Using a 416NHM30032A or SA85002 Modbus Plus network interface adapter, which provides Modbus Plus cable redundancy. This is the most common redundant communications architecture.
- Using two 416NHM30030A or SA85000 Modbus Plus network interface adapters.

Figure 11 Redundant Modbus Plus network architecture with 416NHM30032A or SA85002 adapter**Figure 12** Redundant Modbus Plus network architecture with two 416NHM30030A or SA85000 adapters

Modbus Plus network routing

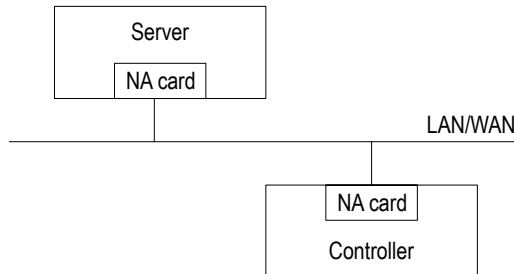
The server supports the five-byte Modbus Plus network routing path. The five bytes allow controllers to be addressed that are up to four networks away. In the following figure, the ID for Controller 2 is 12.7.

Figure 13 A typical Modbus Plus routed connection

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

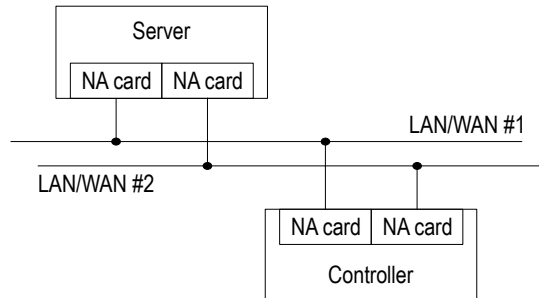
Modbus TCP connection

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.

Figure 14 Non-redundant Modbus TCP network architecture

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.

Figure 15 Redundant Modbus TCP network architecture

Modbus function codes

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

Modbus data tables

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



Note

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 “Reducing the number of data tables being scanned” on page 36.)

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
OP DESTIN	Yes			Yes
MD SOURCE	Yes	Yes	Yes	Yes
MD DESTIN	Yes			Yes
Ax DESTIN	Yes			Yes
Accumulator				
PV SOURCE			Yes	Yes

Setting up a Modbus controller

Setting up a Modbus Plus network

Installing the Modbus Plus adapter

Shut down the server and install the Modbus Plus Adapter as described in the *Modbus IBM Host Based Devices User's Guide*. In addition:

- Ensure the memory base address set via the DIP switches on the adapter is not being used by another adapter. Take note of the base address because this is required when configuring the device driver.
- All other DIP switches should normally be left at the factory default settings. In particular the Polled Mode switch should be set to Polled Mode.

If you are installing a second adapter for a redundant channel, installation is identical to the first adapter except that you must set a different memory base address.

Installing the Modbus software

The Windows NT-based 32-Bit MBX driver for ModConnect Series Host Interface Adapters software comprises a device driver and a set of dynamic link libraries. See the *Modbus ModBus Plus Host Interface Device Driver Manual for Windows NT* for details about installing the software and configuring the device driver.

- The Adapter Number should be set to Device 0.
- Use the memory base address that was set via the DIP switches on the adapter.
- Set the driver to Polled Mode.
- If a second adapter has been installed for a redundant channel the Adapter Number of the second adapter should be set to Device 1.

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 rate (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

3

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

For information about:	Go to:
Defining a Modbus channel	page 20
Defining a Modbus controller	page 26
Defining an address for a point parameter value	page 31
Optimizing scanning performance	page 35

Defining a Modbus channel

To define a channel:

- 1 Click  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 “Modbus channel main properties” on page 20.
- 4 Click the Port tab and for **Port Type** select either:
 - `Serial`
 - `TerminalServer`
 - `LANVendor`

See “Modbus channel port properties” on page 22.

To define a redundant channel:

- 1 Click the Redundant Port tab.
- 2 Select `RedundantSerial` or `LANVendor` as the **Port Type**.
- 3 Set the other properties as appropriate—see “Modbus channel redundant port properties” on page 25.

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

Modbus channel main properties

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

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description (optional)	A description of the channel. A maximum of 30 alphanumeric characters, including spaces.

Property	Description
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. 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 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for 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 $[\sqrt{9} \times 25 = 75]$.
Fail Alarm Limit	The communications alarm 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 in Marginal Alarm Limit.
Diagnostic Scan Rate	The period, in seconds, between diagnostic scans which 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.

Property	Description
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 “Serial port properties” on page 22.
- **terminalserver**. Select this if you want to connect the controller to the server via a LAN. See “TerminalServer port properties” on page 24.
- **lanvendor**. Select this if you want to connect the controller to a Modbus Plus or Modbus TCP network. See “LANVendor port properties” on page 24.



Note

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 Rate	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.

Property	Description
Parity	Set this to EVEN . This setting may be different if you are not using a 584 or 984 Modbus Controller.
Checksum	Set this to None .
XON/XOFF	Set this to None .
RS-232	Only applicable if the controller is connected to the server via an 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.
RS-485	Only applicable if the controllers are connected to the server via an RS-485 link. Select Enable Stallion RS-485 Half Duplex and Echo . (Echo indicates that the server expects messages it sends to the port on the transmit line to be echoed back on the receive line.)
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

TerminalServer port properties

Property	Description
Terminal Server TCP Host Name	The name and port number of the terminal server to which the channel is connected.
Terminal Server TCP Port No	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	Set this to 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. For more information see “Specifying adapter number properties” on page 25. • 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 section “Modbus channel redundant port properties” on page 25 for more information on configuring redundant channels.

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 “Serial port properties” on page 22.



Attention

For Modbus Plus, if you are using a 416NHM30032A or SA85002 Modbus Plus network interface adapter which provides Modbus Plus cable redundancy, you should not configure the channel as redundant, since the redundancy is handled automatically by the adapter. You should only configure a Modbus Plus channel as redundant if you are using two single cable 416NHM30030A or SA85000 adapters

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 page 15) in the physical controller that you need to access. (Unless you have combined data within tables—see “Reducing the number of data tables being scanned” on page 36.)

To define a controller:

- 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 “Modbus controller main properties” on page 26.

Modbus controller main properties

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

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes). For LAN connected controllers, the name must not contain underscore (_) characters. This name is used to look up the IP address in the Hosts file or DNS 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 30 alphanumeric characters, including spaces.
Channel Name	The Modbus channel on which the controller communicates with the server.

Property	Description
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. This limit is the controller barometer, which monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer is incremented by 2 or more, depending on the error and decremented 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. Set this to double the value specified in Marginal Alarm Limit.
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 “Setting up a Modbus controller” on page 17.
Data Table	The data table that this controller represents. The addressable range for point parameter data is always 00001 to 32766, within an individual table. The address ranges for the data tables are as follows: • Digital Output (read/write). 000001 to 032766 • Digital Input (read only). 100001 to 132766 • Input Register (read only). 300001 to 332766 • Holding Register (read/write). 400001 to 432766 Note: When you enter an address, do not include the table number prefix.

Property	Description
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. You can derive the offset by using the formula: $\text{hw address} = 65535 + \text{offset} + \text{specified address}$ For more information on calculating a negative offset, see “Offset and diagnostic address example” on page 29. The diagnostic address determined by the Offset value is overridden if the Diagnostic Address is specified. (See “Offset and diagnostic address example” on page 29.)
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 “Offset and diagnostic address example” on page 29.) If this is not specified, the Offset value is read.
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 controller’s property details.
Last Downloaded	This shows the date that the item was last downloaded to the server.

Property	Description
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.
Item Number	The unique item number currently assigned to this item in the format <code>RTU###</code>. 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.

Offset and diagnostic address example

This example shows the registers used for a diagnostic scan when you specify an offset and diagnostic address.

Offset	Diagnostic Address	First available registers	Register for diagnostic scan
0	0	1	2
50	0	51	52
0	20	1	20 - 21
50	20	51	20 - 21
-30000	0	35536	35537
-30000	200	35536	35735 - 35736

The offset for hardware addresses in the range between 32767 and 65535 are calculated using the formula:

$$\text{HW address} = 65535 + [\text{offset}] + [\text{specified address}]$$

Consider that you had a requirement to build a point that scans the hardware address 40001 and that for diagnostic scans you want to read the registers with the hardware addresses 50000-50001. To allow a specified address of 1 to represent the hardware address 40001, you would work out the offset by solving the following equation: $40001 = 65535 + [\text{offset}] + 1$. Therefore, the offset would be $40001 - 65535 - 1 = -25535$. To work out the diagnostic address to specify you then need to solve the following equation: $50000 = 65535 - 25535 + [\text{specified address}]$. Therefore, the specified address for the diagnostic scan would be $50000 - 65535 + 25535 = 10000$. 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.

Defining a Modbus address for a point parameter

Entering an address

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

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the Modbus controller.
<i>Location</i>	The address within the controller where the value is stored. See “Location syntax” on page 31.

If you would like help when defining an address, click  next to **Address** to display Address Builder. For details, see the help.

Location syntax

The format for the location is:

Address [DataFormat|BitNumber]

Part	Description
<i>Address</i>	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.

Part	Description
<i>DataFormat</i>	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” on page 32 • “Data format for reading floating point values” on page 34 • “Data format for reading raw values without scaling” on page 34 If you want to use a user-defined data format, you must define the format on the server. See the <i>Configuration Guide</i> for details.
<i>BitNumber</i>	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.

Example**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 to 4095
U9999	0 to 9999

Data Format	Counts in Controller Register
U999	0 to 999
U15B	0 to 32767
U16B	0 to 65535
S16B	-32768 to 32767

**Note**

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	Unsigned 32-bit Binary Little-endian	Not scaled
U32B	Unsigned 32-bit Binary Little-endian	Not scaled
S32BS	Signed 32-bit Binary Scaled Little-endian	-32768 to 32767
U32BS	Signed 32-bit Binary Scaled Little-endian	0 to 65535

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	Not scaled
U32BSB	Unsigned 32-bit Binary Scaled Big-endian	0 to 65535
S32BB	Signed 32-bit Binary Big-endian	Not scaled

Data Format	Description	Counts in Controller register
S32BSB	Signed 32-bit Binary Scaled Big-endian	-32768 to 32767

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
IEEEFPPL	Bytes are little endian format
IEEEFPPLB	Bytes are byte-swapped little endian format



Note

- 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 122 *controller register* addresses or 1952 (122*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 (122 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.



Note

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).

Server tasks for Modbus

4

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

For information about:	Go to:
Testing communications with field devices	page 38
Troubleshooting point configuration errors	page 39

Testing Modbus to field device communications

You use the diagnostic utility, **modtst**, to test communications between the Modbus controller and the server.

To run the 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.

Modbus controllers are fitted with transmit and receive diagnostic LEDs. If a communications fault illuminates a LED, see the Modbus manuals in order.

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.

Error code 0106 (Device Timeout)

Probable Cause: No response was received from the controller.

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

Index

A

architecture 8

C

channel, defining 20

communications testing 38

configuring Modbus, steps for 5

connect timeout 21

connection

Modbus Plus 12

RS-422 10

RS-485 11

serial 8

controller definition for a Modbus controller 26

D

defining a Modbus address for a point
parameter 31

devices supported 6

diagnostic scan rate 21

documentation 6

F

function codes 15

M

Main properties for a Modbus channel 20

Main properties for a Modbus controller 26

Modbus

architecture 8

function codes 15

setting up a controller 17

supported devices 6

testing communications 38

troubleshooting 39

Modbus Plus

connection 12

network routing 13

Modbus TCP, connection 14

P

point parameter, defining an address 31

Port properties for a Modbus channel 22

R

read timeout 21

Redundant Port properties for a Modbus
channel 25

RS-422 connection 10

RS-485 connection 11

S

scanning, optimizing performance 35

serial connection 8

serial port properties for a Modbus controller 22

setting up a Modbus controller 17

supported devices 6

T

terminal server properties for a Modbus
controller 24

testing communications 38

timeout

connect 21

read 21

troubleshooting 39

W

write delay 21

Moore APACS Interface Reference

Issue

5

Date

November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for Moore APACS	6
	Moore APACS-specific terms	7
2	Moore APACS controller setup	9
	Connecting to the server	10
	LAN connection	10
	MODULBUS connection	11
	Communication settings	12
	Configuring a LAN connection	12
	Configuring a dual Remote Network Interface	12
	Configuring a MODULBUS connection	13
3	Moore APACS controller configuration and addressing	15
	Defining a Moore APACS channel	16
	Moore APACS channel Main properties	16
	Defining a Moore APACS controller	18
	Moore APACS controller main properties	18
	Defining a Moore APACS address for a point parameter value	20
	Entering an address	20
	Optimizing scanning performance	24
	Moore APACS packets	24
4	Server and Station tasks for Moore APACS	27
	Testing communications with the server	28
	Error messages	28

Getting started

1

This reference describes how to set up, configure, and test Moore APACS controller communications with the server. It also contains detailed information for defining the controller using Quick Builder.

The following table lists the steps for connecting and configuring a Moore APACS controller. Complete each step before commencing the next step.

Step:	Go to:
Set the communication parameters	page 12
Connect the cables from the server to the Moore APACS controller	page 10
Define channels using Quick Builder	page 16
Define controllers using Quick Builder	page 18
Download channel and controller definitions to the server	
Test communications	page 28
Troubleshooting communication errors	page 28
Define points with Quick Builder	page 20

Support and documentation for Moore APACS

For information about supported devices and related documentation, see the Software Change Notice or the Software Release Bulletin.

Moore APACS-specific terms

APACS

Advanced Process Automation and Control System

DataTag

Process parameter or variable configured in the APACS controller. Properties include name, type, quality and value.

DCS

Distributed Control System

IEM

Industrial Ethernet Module (see RNI).

MBI

MODULBUS Interface card that is installed in the server. Required to connect the controller to the server via MODULBUS.

M-BUS, MODULBUS

Proprietary Moore communications protocol used by APACS.

NIM

Network Interface Module (see RNI)

RBE

Report By Exception

RNI

Remote Network Interface. Required to connect the controller to the server via LAN.

Moore APACS controller setup

2

This chapter describes how to set up an Moore APACS controller.

For details about:	Go to:
Connecting to the server	page 10
Communication settings	page 12

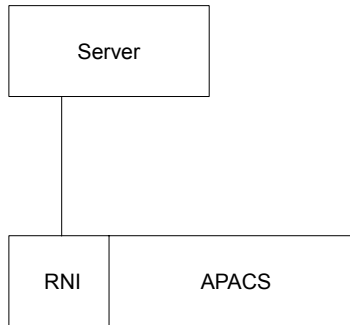
Connecting to the server

You can connect the Moore APACS controller to the server using either LAN or MODULBUS communications. It is not possible for a single server to be configured with both LAN and MODULBUS channels.

LAN connection

If you connect the Moore APACS controller to the server with an Ethernet connection, you must have a Remote Network Interface (RNI) module installed.

Figure 1 LAN connection

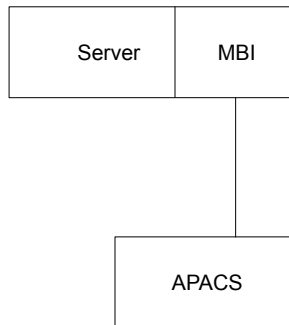


MODULBUS connection

If you connect the Moore APACS controller to the server with a MODULBUS connection, you must have a MODULBUS Interface (MBI) card installed.

See the Moore documentation for potential issues with the MBI card. For example, on Dell PCs with a custom Dell BIOS, it may not be possible to reserve an area of shadow RAM for the MBI card. It may be possible to use a Dell computer without reserving the shadow RAM, but this could change on a computer by computer basis.

Figure 2 MODULBUS connection



Communication settings

After connecting the Moore APACS controller to the server, configure the communications settings.



Note

The controller's communication settings are in the file `c:\etc\apacs.ini`.

Configuring a LAN connection

To configure a LAN connection:

- 1 Connect the server to the Ethernet network with the RNI.
- 2 Install the APACS drivers/API software, selecting to install the **Ethernet API DLL files**.
- 3 If prompted to **Select version of Runtime API to copy**, select **Ethernet API DLL files**.
- 4 Reboot when prompted.
- 5 Ensure the IP address of the Remote Network Interface (RNI) is defined in the TCP/IP host file on the server, `c:\windows\system32\drivers\etc\hosts`.
For example: `nim2 10.11.25.2`
- 6 Ensure that the name of the server and its IP address are in the RNI's host file. It may be necessary to manually edit the RNI's host file rather than using the standard RNI software.
For example: `servera 10.11.25.57`
- 7 Record the name of the RNI because you need it when using Quick Builder – see “Defining a Moore APACS channel” on page 16.

Configuring a dual Remote Network Interface

You can configure dual Remote Network Interfaces (RNIs) in the Moore APACS channel configuration RNI by separating the names with a semi-colon; for example:

```
apacsnim1;apacsnim2
```

The APACS channel will use either of these for communication with the Moore APACS system.

Expected channel interruptions

The dual-RNI scenario should not be considered a fault-tolerant solution. If either of the RNIs configured for the channel is disconnected or reconnected, interruptions may occur on the channel (typically a pause of 15 seconds after a single disconnection/reconnection event). If multiple disconnection/reconnection events occur within a 5-minute period, multiple pauses, bad point values, and channel failure may all occur.

Diagnostics

To configure diagnostic information for the dual-RNI channel:

- 1 In 4-mation, build a repeat cycle timer (function block REPCYCL) in the APACS resource to cycle between 0 and 60 seconds as an integer.
- 2 In Quick Builder:
 - a. Build an extra APACS channel/controller for each individual RNI. (This will be used for dual-RNI diagnostics only.)
 - b. Build a point on each diagnostic controller with the PV addressing the timer integer variable.
- 3 In Station, a communication alarm will be raised on the diagnostic channel/controller if the RNI is unavailable.

The value of each timer point can be monitored to verify successful data communication with the corresponding RNI.

Configuring a MODULBUS connection

To configure a MODULBUS connection:

- 1 Set the interrupt level on the MBI card to a free interrupt on the server (default IRQ 10).
- 2 Install the card in the server.
- 3 Install the APACS drivers/API software and ensure the driver starts correctly before continuing.
- 4 When using Quick Builder to configure the APACS channel, set the RNI name to **MBI** - see “Defining a Moore APACS channel” on page 16.

Moore APACS controller configuration and addressing

3

This chapter describes how to configure a Moore APACS controller using Quick Builder. For each configuration procedure, there is detailed information covering all supported Moore APACS models.

For details about:	Go to:
Defining a Moore APACS channel	page 16
Defining a Moore APACS controller	page 18
Defining an address for a point parameter value	page 20
Optimizing scanning performance	page 24

Defining a Moore APACS channel

You need to define a channel for the connection the server uses to communicate with Moore APACS controllers.

To define a channel using Quick Builder:

- 1 Click  to add a channel.
- 2 In the Add Items dialog box, select `channel` as the item and `Moore APACS` as the channel type.
- 3 Enter the channel details on the Main property page for the channel. For a description of the properties, see Moore APACS channel Main properties.

Moore APACS channel Main properties

The Main tab defines the basic properties for a Moore APACS channel.

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description	(Optional) A description of the channel. 30 characters maximum, 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. 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 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for 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 $[3 \text{ is square root }] \times 25 = 75$.
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. Set this to double the value specified for the channel Marginal Alarm Limit.

Property	Description
Background Read Interval	The period at which a full read is done of all configured addresses. At other times, only values which have changed are read (see “Moore APACS packets” on page 24). This should be set to a reasonable value so it does not impact performance, for example, 300.
Remote Network Interface Name	The name of the RNI to which the channel is connected. If the channel is connected directly to the MODULBUS, this should be set to “MBI”.
Item Type	Shows the channel type.
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. 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.

Defining a Moore APACS controller

Moore APACS controllers are used as logical collections of APACS tags, rather than physical APACS devices. A controller definition can contain up to 40 characters of an APACS DataTag name, which is prepended to the point addresses. See “Defining a Moore APACS address for a point parameter value” on page 20.

You need to define a controller to describe each Moore APACS on a channel.

To define a controller using Quick Builder:

- 1 Click  to add a controller.
- 2 In the Add Items dialog box, select Controller as the item and Moore APACS as the controller type.
- 3 Enter the controller details on the Main property page for the controller.

Moore APACS controller main properties

The Main tab defines the basic properties for a Moore APACS controller.

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description	A description for this controller. A maximum of 30 characters, including spaces.
Channel Name	The name of the channel on which the controller communicates. You need to have already defined the channel in order for it's name to appear in the list.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. The default value is 25.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this value is reached, an urgent alarm is generated. Set this to double the value specified for the controller Marginal Alarm Limit.
Base DataTag	(Optional) A Base DataTag Name that will be prepended to all points built on this controller. “Defining a Moore APACS address for a point parameter value” on page 20.
Item Type	Shows the controller’s type.
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. 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.

Defining a Moore APACS address for a point parameter value

Entering an address

For source and destination addresses, the format for a Moore APACS controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the Moore APACS controller.
<i>Location</i>	The location in the controller where the value is recorded. See Location syntax.

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

Location syntax

The format for the location is:

Address [DataFormat | BitNumber]

Part	Description
<i>Address</i>	The address of the point parameter. See “DataTag name generation” on page 21. This can have a maximum size of 64 characters.
<i>DataFormat</i>	The data format acronym of the point parameter. See “Data format” on page 21.
<i>BitNumber</i>	The starting bit number for the point parameter that references a register in the APACS. The valid range is 0 (default) to 15 where 0 is the right-most bit in the register. The number of bits read is implied by the parameter type: analog parameters read up to 16 bits, status parameters read 1, 2 or 3 consecutive bits.

DataTag name generation

The full DataTag name for a point consists of the controller's Base DataTag plus the address configured in the point address definition.

[BaseDataTag.]PointAddress

For example, if many points are going to be defined from a certain program, the Base DataTag could be configured as *ResourceName.ProgramName*. Server points could then be defined using *FunctionBlock.Parameter* or *ServerVariable*.

Example

To build points on DataTags:

`Reactor.Area2.Unit2.TC1100.PV`

`Reactor.Area2.Unit2.SteamTemp`

Use the following:

Item	Value
Controller <i>Base DataTag</i>	Reactor.Area2.Unit2
Point1 <i>Address</i>	TC1100.PV
Point2 <i>Address</i>	SteamTemp

Data format

The following standard data formats can be used with Moore APACS points. (If you want to use a user-defined data format, you must define the format on the server. See the *Configuration Guide* for details.)

Data Format	Range	Scaled by Server Range
<none>		No
S8B	-128 to 127	Yes
S16B	-32768 to 32767	Yes
S32B	-2147483648 to 2147483647	No
U8B	0 to 255	Yes
U16B	0 to 65535	Yes
U32B	0 to 4294967296	No
IEEEFP	single precision floating point	No

**Note**

- It is not necessary to specify the data type unless you want the value to be scaled by the point range. The data type property of the APACS DataTag will be used to read the value in an appropriate unscaled format.
- The date and time related APACS addresses will be read as a 4-byte unsigned integer value. The server will not manipulate this value, but it is available for use in Excel Data Exchange or a custom application.

Array DataTags

Array DataTags in the Moore APACS controller are accessed as individual array elements. For information about Array DataTags, see the *Moore Configuration Guide – API Toolkit Version 4.00 or higher*.

**Note**

The APACS supports multi-dimensional arrays, which are accessed by address [x,y,z]. Because Quick Builder does not support commas in address definitions, use colons. This is converted to a comma internally.

Example

To build points on elements 1 and 4 of Array DataTag `SteamTemps`, use the following addresses:

```
Reactor.Area2.Unit2.SteamTemps[1]  
Reactor.Area2.Unit2.SteamTemps[4]
```

To build points on elements 1,2 and 4,1 of Array DataTag `SteamTemps`, use the following addresses:

```
Reactor.Area2.Unit2.SteamTemps[1:2]  
Reactor.Area2.Unit2.SteamTemps[4:1]
```

Global variables

Global variables appear on the 4-mation screens with square brackets around the variable name. The correct TagName for global variables is `<ResourceName>.<VariableName>`. The variable should not be addressed as an item on the page in 4-mation.

Example

On the page `Reactor.Area2.Unit2`, a variable appears as `[ShutDown]`. To build a point to reference this value, use `Reactor.ShutDown`.

MODE addressing

The mode setting of control loops in the APACS is stored in two addresses. A multiplexed address is used for the server MODEs to read these. The format of this multiplexed address is as follows:

```
Reactor.Area2.Unit2.SteamTemp!AUTO!EXTRNL
```

This address is split into two addresses using the exclamation marks (!) as delimiters. The following addresses would be read from the controller:

```
Reactor.Area2.Unit2.SteamTemp.AUTO
```

```
Reactor.Area2.Unit2.SteamTemp.EXTRNL
```

The first address is mapped to the Auto/Manual property of the server MODE, and the second address is mapped to the Local/Remote property.

Even if the MODE is only being built to read the Auto/Manual parameter, you can still use the ! notation.

```
Reactor.Area2.Unit2.SteamTemp!AUTO
```

Multiplexed status points

Status points can also be built using the ! notation. The point uses the first address as the bit0 value, and the second address as the bit1 value. Each address used must return a 0/1 value.

For example: `Reactor.Area2.Unit2.SteamValve!OP_VER!CL_VER`

Optimizing scanning performance

Minimize the number of scan packets, and the burden on the server, by using a small number of available scan periods for all your point definitions.

The scan packets that have been built can be listed by using the list scan utility, **lissen**. Listing scan packets helps verify the scanning strategy. See the *Configuration Guide* for your server for details about **lissen**.

The following are examples of scan periods:

- 5 seconds for fast changing, important parameters
- 60 seconds for 1 minute PV history, parameters that change every few minutes
- 300 seconds for slow changing parameters



Note

Performance of >3000 addresses per second was observed during testing.

Moore APACS packets

A Moore APACS scan packet can consist of up to 100 APACS tags and these tag addresses can be located anywhere in the APACS system.

Three types of scan packets are built for a Moore APACS interface:

- Hardware diagnostic scan
- Periodic data acquisition scan
- Full acquisition scan

Hardware diagnostic scan

One scan per controller every 60 seconds is initiated automatically by the server to verify communications integrity with the controller. One hardware diagnostic scan is automatically created per controller.

Periodic data acquisition scan

Periodically, the server acquires values from a controller and processes these values as point parameters. For each source address scanned, a scan period can be specified. The period specified depends on the nature of the parameter. A fast changing or critical parameter uses a fast rate; an infrequent or non-critical (for example, SP) parameter should use a slower rate. For periodic data scanning, the server uses functions of the Moore API to only update those addresses whose values have changed since the last scan.

Full acquisition scan

Periodically, the server reads all source addresses that have been configured. Every point parameter using these source addresses has its value updated.

Server and Station tasks for Moore APACS

4

This chapter describes tasks for the Moore APACS controller that you perform either on the server or from any Station.

For details about:	Go to:
Testing communications with the server	page 28

Testing communications with the server

You use the Moore APACS test utility, **apactst**, to test communications between the server and the controller after you have downloaded your channel and controller definitions to the server database.

Before testing:

- 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

To run the utility:

- 1 Open a Command Prompt window.
- 2 Type **apactst** and press ENTER.
- 3 Follow the directions as prompted.

Error messages

APACTST – Unable to Locate DLL

Install the Moore Runtime API v4.31 on the server.

InitializeDataServices() Error: Invalid NIM Name

Ensure that the configured NIM name is in the server's hosts file.

ReadTagList() Error: No modules responded

Check the communications link between the server and the RNI.

ReadTagList() Error: Tag Name is invalid

Check the tag name in the APACS.

“GetIOStatus(<TagName>) Error: Tag Doesn't Exist”

The TagName configured does not exist in the APACS. Check the TagName against the Moore 4-Mation software to verify the address. This error message can occur when the TagName references an input pin of a block that does not have a wire or variable attached to it.

“GetDataQuality(<TagName>) Error: data marked as <DataQuality>”

This tag has a bad value in the APACS.

“ValidateTagName(<TagName>) Error: Unknown Error”

This occurs when a global variable is addressed incorrectly. See “Global variables” on page 22.

“WriteBit(<TagName>) Error: Invalid Request”

A status point has been built on an APACS address that does not allow individual bit control. Use an analog point to control this address.

“Caught access violation when processing tag <TagNumber> (<TagName>) - setting quality to bad”

Typically for TagNames that have previously displayed GetIOStatus() or GetDataQuality() errors. No user action required.

Index

C

- channel properties
 - Main tab 16
- channel, defining 16
- communication settings 12
- communications, testing 28
- configuring Moore APACS, steps for 5
- controller properties
 - Main tab 18
- controller, defining 18

M

- Moore APACS
 - channel, defining 16
 - communication settings 12
 - configuration steps 5
 - connecting to the server 10
 - controller, defining 18
 - point parameter address 20
 - testing communications 28

P

- point parameter address, defining 20
- properties
 - channel 16
 - controller 18

S

- scanning, optimizing performance 24
- server, connecting to 10

Moore Mycro Interface Reference

Issue

5.1

Date

November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for Moore Mycro	6
	Moore Mycro models supported	6
	Other documentation for Moore Mycro	6
2	Moore Mycro controller setup	7
	Supported architecture.	8
	Connecting Moore Mycro 352 to the server	9
	Communication settings	11
3	Moore Mycro controller configuration and addressing	13
	Defining a Moore Mycro channel	14
	Moore Mycro channel Main properties	14
	Moore Mycro port properties	15
	Defining a Moore Mycro controller	18
	Moore Mycro controller main properties	18
	Defining a Moore Mycro address for a point parameter value	20
	Entering an address	20
	Addresses supported by the interface	23
	Optimizing scanning performance	40
	Moore Mycro scan packets	40
4	Server and Station tasks for Moore Mycro	41
	Testing communications with the server	42
	Troubleshooting point configuration errors.	43

Getting started

1

This reference describes how to set up, configure, and test Moore Mycro controller communications with the server. There is detailed information for defining the controller using Quick Builder.

Complete each step before commencing the next.

Steps for connecting and configuring a Moore Mycro 352 controller:

Steps	Go to:
Set the communication parameters	page 11
Connect to the server via RS-232 or RS-422	page 9
Use Quick Builder to define channels:	page 14
Use Quick Builder to define controllers	page 18
Download channel and controller definitions to the server	
Test communications	page 42
Define points with Quick Builder	page 20

Support and documentation for Moore Mycro

Moore Mycro models supported

The server specifically supports the Moore Mycro 352 device using this interface. See the *SLDC Link Interface Communications User's Manual* for detail on available addresses for this device.

While the interface has not been tested with the following devices, the communication protocol used is the same as the Moore Mycro 352 device. This interface should therefore work with the following controllers:

- Moore Model 324 Programmable Sequence Controller
- Moore Model 351 Triple Loop Digital Controller
- Moore Model 382 Logic and Sequence Controller
- Moore Model 383 Multi-Point Display Station

Other documentation for Moore Mycro

The following Moore Products books describe the address space of all the devices supported by this interface in great detail. They also provide information about the physical configuration of the Units. Reading these books would be useful when installing the interface.

- *Model 352 SLDC Link Interface Communications User's Manual*; Part No: AD352-40
- *Model 352 SLDC Installation and Service Instruction*; Part No: SD352

The manuals listed below provide detailed information concerning the configuration of other units.

- *Model 320 Local Instrument Link Independent Computer Interface User's Manual*; Part No: AD320-10
- *Local Instrument Link - Installation & Service Manual*; Part No: SD15492

Moore Mycro controller setup

2

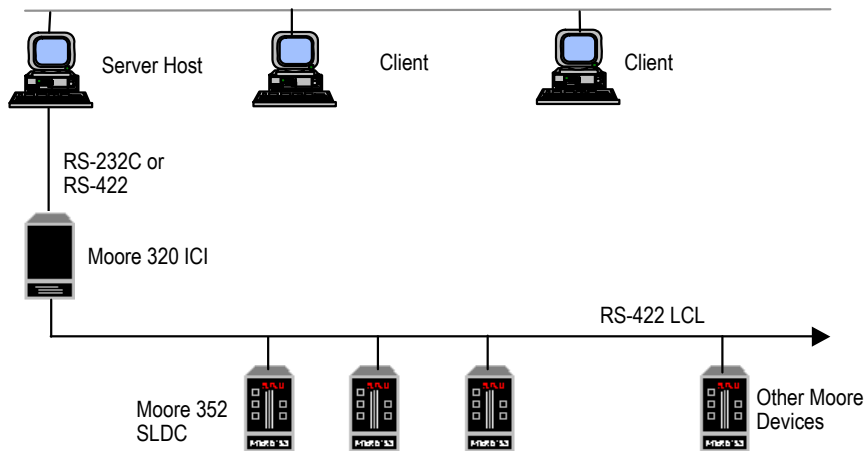
This chapter describes how to set up an Moore Mycro controller for the system.
The tasks for setting up a Moore Mycro controller are:

For:	Go to:
Supported architectures	page 8
Connecting to the server	page 9
Communication settings	page 11

Supported architecture

An example of a system design integrating the server with a network of Moore Controllers.

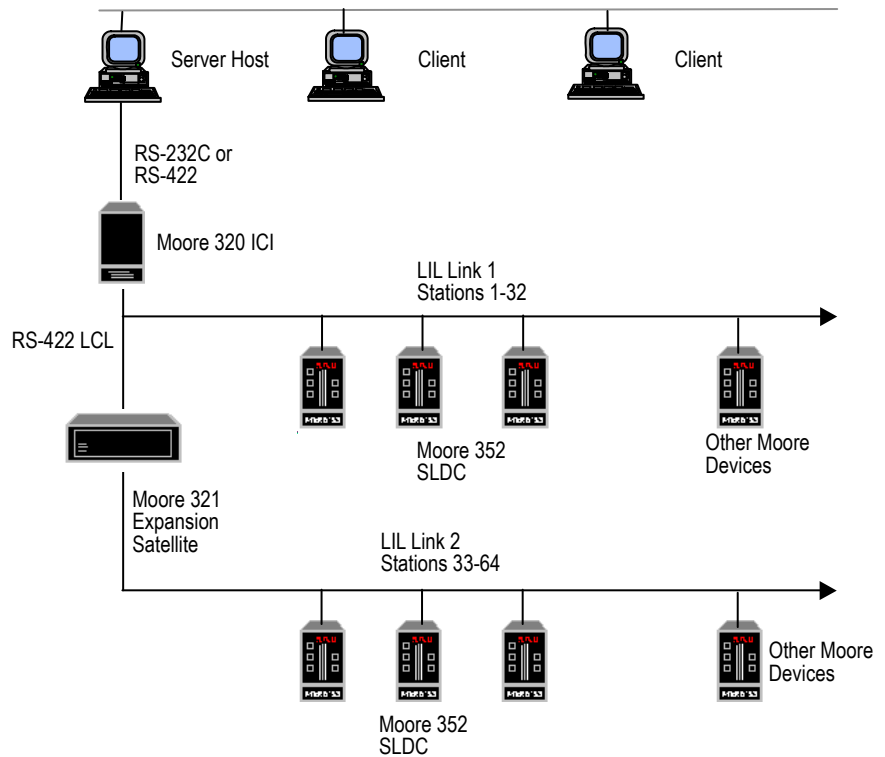
Figure 1 Sample system design—minimum system



The RS-232 and RS-422 server interface to the Moore 320 ICI allows only one connected ICI unit per server channel.

As many as 32 Moore controllers can be connected at once on a single Local Instrument Link (LIL).

Using two links enables up to 64 Moore stations to be connected to a single channel. “Sample system design—expanded system” on page 9 describes an expanded LIL system.

Figure 2 Sample system design—expanded system

Connecting Moore Mycro 352 to the server

Configure the Moore Mycro 352 SLDC and Moore Mycro 320 ICI before connecting to the server. The following configuration must be performed to allow the server interface to communicate correctly with the devices. See the Moore Mycro documentation for details on setting up the following configuration.

- 1 Enable the Link Interface Option Function Block on the Moore Mycro 352 (Function Block 98 on the device), then set up the relevant internal variables in Function Block 98 to point to the appropriate internal variables.
- 2 Ensure that the Moore is in Console Source mode (as opposed to Computer Source mode)—this is verified by building a status point to read Bit 4 of LIST (see “Addresses supported by the interface” on page 23).
- 3 The Moore 320 ICI must be set up to use Binary Transmission Mode, Link Acknowledge Delay Enabled, and Null Filled Data Enabled.

Any other configuration of the Moore PLCs should be done by direct hardware configuration, a third-party loader package, or if possible by using the Moore Mycro test utility, **mmutst**. For instructions on running the test utility, see “Testing communications with the server” on page 42.

Communication settings

Set up the communication parameters between the server and the Moore ICI. See the Moore documentation for details on setting up these parameters.

The Unit Address of each device on the Local Instrument Link must be set to a unique value between 1 to 64.

Parity is selectable for the interface between the ICI and the server computer. Parity may be ODD, EVEN or NONE. Use of parity is recommended.

Baud Rates are selectable for the interface between the ICI and the server computer. The following table displays selectable baud rates for the Moore ICI:

BAUD Rate	Moore 320 ICI
38400	Yes
19200	Yes
9600	Yes
4800	Yes
2400	Yes
1200	Yes
300	Yes

Moore Mycro controller configuration and addressing

3

This chapter describes how to configure a Moore Mycro controller for the system using Quick Builder. For each configuration procedure, there is detailed information covering all supported Moore Mycro devices.

The Quick Builder controller configuration tasks are described:

For:	Go to:
Steps for defining a Moore Mycro channel	page 14
Steps for defining a Moore Mycro controller	page 18
Defining an address for a point parameter value	page 20
How to optimize scanning performance	page 40

Defining a Moore Mycro channel

You need to define one server channel for each connection with a Moore Mycro 320 ICI.

To define a channel using Quick Builder:

- 1 Click  to add a channel.
- 2 In the Add Items dialog box, select `channel` as the item and `Moore Mycro` as the channel type.
- 3 Enter the channel details on the Main property page for the channel. For help with the channel definitions, see Moore Mycro channel Main properties.
- 4 To complete the channel definition, click the Port tab and define a port. See “Moore Mycro port properties” on page 15.

Moore Mycro channel Main properties

Use the Main tab to enter the basic channel properties for a Moore Mycro channel.

Property	Description
Name	Type a unique name for the channel. A maximum of 10 alphanumeric characters can be used.(Double quotation marks and spaces are not allowed.)
Description	(Optional) Type a description of the channel. A maximum of 30 characters can be used, 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. 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 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for 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 10, the value would be [3 is square root] x 10 = 30.

Property	Description
Fail Alarm Limit	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.
Connect Timeout	Amount of time (in seconds) the server waits to connect to the controller before abandoning the connection. Type a new value in this field if you do not want the default, 10 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Read Timeout	Amount of time (in seconds) the server waits for a reply from the controller. Type a new value in this field if you do not want the default, 2 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
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.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

Moore Mycro port properties

The Port tab defines the communication-related properties for a channel. The only Port Type you can define for a Moore Mycro is `Serial`.



Note

The Serial Port settings must match the settings on your communication devices.

Serial port properties

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	The number of data bits per second. The default is 9600.
Number of Data Bits	The number of data bits used for transmission. The default is 8.
Stop Bits	The number of stop bits used for transmission. The default is 1.
Parity	The parity verification used on the port. The default is NONE.
Checksum	The type of checksum error detection used for the port. Select the value that matches the setting on the communication device: • CRC16_0 or CRC16_1 (if Cyclic Redundancy Check (CRC) is set) • ONESCOMP or TWOSCOMP (if Longitudinal Redundancy Check (LRC) is set) • XOR (If exclusive or is set)
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: • None (default) • Input (use XON/XOFF to control the flow of data on the receive line) • output (use XON/XOFF to control the flow of data on the transmit line)

Property	Description
Handshaking Options	RS-232 • Enable RTS/CTS flow control. Select 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 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 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. RS-422. No options available. RS-485. Not applicable for Moore Mycro.

Defining a Moore Mycro controller

You need to define each Moore station as a controller on a server channel.

To define a controller using Quick Builder:

- 1 Click  to add a controller.
- 2 In the Add Items dialog box, select Controller as the item and Moore Mycro as the controller type.
- 3 Enter the property definitions for the controller on the controller Main property page.

Moore Mycro controller main properties

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

Property	Description
Name	Type a unique name for the controller. A maximum of 10 alphanumeric characters can be used. (Double quotation marks and spaces are not allowed.)
Description	(Optional) Type a meaningful description for this controller. A maximum of 30 characters, including spaces, can be entered.
Channel Name	The name of the channel on which the controller communicates. In the list of channel names, click the name. You need to have defined the channel in order for it's name to appear in the list.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. Type a new value in this field if you do not want the default, 25.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this value is reached, an urgent alarm is generated. To determine an acceptable limit, double the Marginal Alarm Limit value specified for this controller.
Controller Type	Select the controller type as one of: • 352 Single Loop Digital Controller • 320 Independent Computer Interface • 383 Multi Point Display Station • 351 Triple Loop Digital Controller • 324 Programmable Sequence Controller • 382 Logic and Sequence Controller
Station Address	The station address (LSA) of the Moore Mycro device.
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.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field 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.

Defining a Moore Mycro address for a point parameter value

Entering an address

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

```
ControllerName Location
```

Part	Description
ControllerName	The name of the Moore Mycro controller.
Location	The location in the controller where the value is recorded. Location syntax will depend on the type of address format you select: • Address Format Option 1. The preferred form of addressing. See “Addressing—Option 1” on page 20. • Address Format Option 2. Explicitly states the channel and parameter number of the parameter required. It should be used only to specify addresses on the device which are not listed in “Addresses supported by the interface” on page 23 or addresses where no mnemonic has been defined. See “Addressing—Option 2” on page 21.

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

Addressing—Option 1

To specify the location of a point parameter in a controller LPU or other controller, the preferred address format is:

```
A:Mnemonic [B:bb|ParamType]
```

Part	Description
Mnemonic	ASCII representation of the address location (see “Addresses supported by the interface” on page 23).

Part	Description
<i>bb</i>	(Optional) Bit Number for status point addresses only. Value between 0 and 15. If specified, only the raw bit value is used. Cannot be used in conjunction with ParamType. All status point types in server and MD parameters of a point are automatically set to address bits (Status Word type) even if the B: command line is not specified. If the B: is not specified for a status word then the bit offset is set to the default of 0.
<i>ParamType</i>	(Optional) Tuning parameter type override. If specified, will override the parameter type given in the device address definition file, <code>data/mmu_352_def</code> . Cannot be used in conjunction with bit number. See “Parameter data types” on page 22.

Example

A:L1S (Loop 1 Setpoint)

A:L2V (Loop 2 Valve)

A:SS B:6 (Configuration Hold Mode of Station Status Word)

A:ING DIG (Channel G as a Digital)

Addressing—Option 2

While the Moore Mycro 352 Interface should work with all addressable parameters, it has not been tested with all of them and its operation is not guaranteed.

The addressing format is:

`C:ccc P:ppp [B:bb|ParamType]`

Part	Description
<i>ccc</i>	Moore Channel (not the server channel) in the Mycro 352, between 1 and 256.
<i>ppp</i>	Moore Parameter, between 1 and 256.

Part	Description
<i>bb</i>	(Optional) Bit Number for status point addresses only. Value between 0 and 15. If specified, only the raw bit value is used. Cannot be used in conjunction with ParamType. All status point types in server and MD parameters of a point are automatically set to address bits (Status Word type) even if the B: command line is not specified. If the B: is not specified for a status word then the bit offset is set to the default of 0.
<i>ParamType</i>	(Optional) Tuning parameter type override. If specified, will override the parameter type given in the device address definition file, <code>data/mmu_352_def</code> . Cannot be used in conjunction with bit number. See “Parameter data types” on page 22.

**Note**

The channel or the parameter can be entered in decimal, hexadecimal or octal. If the channel or parameter is prefixed by 0x, then it is processed as hexadecimal, for example, C:0x10 P:0x08 = C:16 P:8. If the channel or parameter is prefixed by 0 then it is processed as octal, for example, C:010 P:005 = C:8 P:5. If there is no prefix it is treated as decimal.

Example

C:6 P:3 (Time Integral for Moore 352 Controller #1 with Time scaling)

C:19 P:1 (Channel B Alarm Status word)

C:3 P:1 B:14 (Bit of the status parameter which indicates whether the associated SLDC Station has an error)

C:210 P:20 PRCNT (A previously undefined address at channel 210 parameter 20)

Parameter data types

There are eight different parameter data types that can be applied to each point address.

Data Type	Description	Counts	Server Scaled
PGAIN	Proportional Gain	-100.0 to 100.0	No
ITIME	Integral Time	0.01 to 1000	No
DTIME	Derivative Time	0.00 to 100.0	No

Data Type	Description	Counts	Server Scaled
DGAIN	Derivative Gain	1.00 to 30.00	No
PRCNT	Percentage (Default)	0 to 100.0%	No
STAND	Standard (Scaled)	0 to 100.0%	Yes
C16	One to one mapping	0 to 65535	No
DIG	Mapped Digital	0 or 1	No

The parameter data type STAND is a scaled format of PRCNT and will be scaled by server to a value between the ranges set by the RANGE keyword during the point build. Certain addresses on the Moore 352 device require specific RANGE settings to output the correct value. See “Addresses supported by the interface” on page 23 for details.

All the parameter data types are 16-bit (2-byte) words. The default data type for predefined parameters is specified in the device address definition file `mmu_352_def`. The default parameter data type for addresses which are not defined in the definition file is the Percentage Type. The device address definition file also uses the ASCII string STATS to define addresses which are Status Words.

The parameter data type DIG allows Moore digital addresses to be interpreted as a digital value in the server. The Moore 352 internally stores a digital value as an analog value. It uses the value 0x080 for the value 0 and the value 0xF80 for the value 1. Setting a status point parameter’s data type to be DIG causes the server to automatically perform this mapping when reading or writing from a Moore 352 address.

Example

Configure a status point PV address as A:IND DIG

The server will read the analog value at the address IND. If the value is 0xF80 then the status point will be set to 1. If the value is 0x080 then the status point will be set to 0.

Addresses supported by the interface

All the Moore channel parameters listed in the *Moore Model 352 SLDC User’s Manual* are addressable. Note that not all parameters are able to be controlled.

The ASCII names for each address are defined in text files found in `server\data`. There is one text file for each type of Moore Mycro device. To add additional ASCII names for addresses or to look up the configured ASCII name for an address, view the associated device address definition file for that device.

Moore Mycro Device	Device Address Definition Filename
Moore 352	mmu_352_def
Moore 320	mmu_320_def
Moore 383	mmu_383_def
Moore 351	mmu_351_def
Moore 324	mmu_324_def
Moore 382	mmu_382_def

Related Topics

“Global data parameters (Parameter 1) supported for the Moore 352” on page 24.

“Data parameters other than 1 supported for the Moore 352” on page 26.

Global data parameters (Parameter 1) supported for the Moore 352

Check the Read/Write column to determine which addresses are writable. The interface does not allow any of the database records to be accessed.

Channel	Param Type	Mnemonic	Read/Write	Description
1	C16	SDS	R	Station Data Size (\$0026)
2	C16	ST	R	Station Type (\$0001)
3	Status word 4	SS	R/W	Station Status
4	C16	SEC	R	Station Error Code
5	Status word 5	FBS	R/W	Function Block Status
6	STAND	L1P	R	Loop 1 Process ⁱ
7	STAND	L1S	R/W	Loop 1 Setpoint
8	PRCNT	L1V	R/W	Loop 1 Valve
9	Status word 2	L1ST	R/W	Loop 1 Status
10	Status word 6	L1A	R/W	Loop 1 Alarm Status
11	STAND	L2P	R	Loop 2 Process
12	STAND	L2S	R/W	Loop 2 Setpoint

Channel	Param Type	Mnemonic	Read/Write	Description
13	PRCNT	L2V	R	Loop 2 Valve
14	Status word 15	L2ST	R/W	Loop 2 Status
15	STAND	INA	R	Channel A
16	Status word 7	CAA	R/W	Channel A Alarm Status
17	Status word 8	CAS	R/W	Channel A Status
18	STAND	INB	R	Channel B
19	Status word 9	CBA	R/W	Channel B Alarm Status
20	Status word 10	CBS	R/W	Channel B Status
21	STAND	INC	R	Channel C
22	Status word 11	CCA	R/W	Channel C Alarm Status
23	Status word 12	CCS	R/W	Channel C Status
24	STAND	IND	R	Channel D
25	Status word 13	CDA	R/W	Channel D Alarm Status
26	Status word 14	CDS	R/W	Channel D Status
27	STAND	INE	R	Channel E
28	STAND	INF	R	Channel F
29	STAND	ING	R	Channel G
30	STAND	INH	R	Channel H
31	STAND	O70	R/W	FB98 Output No. 70
32	STAND	O71	R/W	FB98 Output No. 71
33	STAND	O72	R/W	FB98 Output No. 72
34	STAND	O73	R/W	FB98 Output No. 73
35	STAND	O74	R/W	FB98 Output No. 74
36	STAND	O75	R/W	FB98 Output No. 75
37	STAND	O76	R/W	FB98 Output No. 76
38	STAND	O77	R/W	FB98 Output No. 77

- i For L1P, L1S, L2P, L2S - RANGE in Quick Builder should match the Process HI and LO configured on the device

Data parameters other than 1 supported for the Moore 352

Channel	Parameter	Param Type	Mnemonic	Read/Write	Description
6	2	PGAIN	PG1	R/W	Proportional Gain for Controller #1
6	3	ITIME	TI1	R/W	Time Integral for Controller #1
6	4	DTIME	TD1	R/W	Time Derivative for Controller #1
6	5	DGAIN	DG1	R/W	Derivative Gain for Controller #1
6	6	PRCNT	MR1	R	Manual Reset for Controller #1
6	7	STAND	R	R	Ratio for FB07 ⁱ
6	8	STAND	B	R	Bias for FB08 ⁱⁱ
7	2	PRCNT	TSP1	R/W	Target Setpoint for Setpoint #1
7	3	STAND	RT1	R/W	Ramp Time for Setpoint #1 ⁱⁱⁱ
7	4	PRCNT	SPHL1	R/W	Hi Limit setting of HI/LO Limit #1
7	5	PRCNT	SPLL1	R/W	Lo Limit setting of HI/LO Limit #1
10	2	PRCNT	L1A1L	R/W	Loop 1 Alarm #1 Limit
10	3	PRCNT	L1A2L	R/W	Loop 1 Alarm #2 Limit
10	4	PRCNT	L1A3L	R/W	Loop 1 Alarm #3 Limit
10	5	PRCNT	L1A4L	R/W	Loop 1 Alarm #4 Limit
10	6	Status	L1A1T	R	Loop 1 Alarm #1 Type Word
10	7	Status	L1A2T	R	Loop 1 Alarm #2 Type Word
10	8	Status	L1A3T	R	Loop 1 Alarm #3 Type Word
10	9	Status	L1A4T	R	Loop 1 Alarm #4 Type Word
11	2	PGAIN	PG2	R/W	Proportional Gain for Controller #2
11	3	ITIME	TI2	R/W	Time Integral for Controller #2

Channel	Parameter	Param Type	Mnemonic	Read/Write	Description
11	4	DTIME	TD2	R/W	Time Derivative for Controller #2
11	5	DGAIN	DG2	R/W	Derivative Gain for Controller #2
11	6	PRCNT	MR2	R/W	Manual Reset for Controller #2
11	7	ITIME	TH	R/W	Time Lead for FB41
11	8	ITIME	TL	R/W	Time Lag for FB40
12	2	PRCNT	TSP2	R/W	Target Setpoint for Setpoint #2
12	3	STAND	RT2	R/W	Ramp Time for Setpoint #2 ^{IV}
12	4	PRCNT	SPHL2	R/W	Hi Limit setting for HI/LO Limit #2
12	5	PRCNT	SPLL2	R/W	Lo Limit setting for HI/LO Limit #2
16	2	PRCNT	CAA1L	R/W	Channel A Alarm #1 Limit
16	3	PRCNT	CAA2L	R/W	Channel A Alarm #2 Limit
16	6	Status	CAA1T	R	Channel A Alarm #1 Type Word
16	7	Status	CAA2T	R	Channel A Alarm #2 Type Word
19	2	PRCNT	CBA1L	R/W	Channel B Alarm #1 Limit
19	3	PRCNT	CBA2L	R/W	Channel B Alarm #2 Limit
19	6	Status	CBA1T	R	Channel B Alarm #1 Type Word
19	7	Status	CBA2T	R	Channel B Alarm #2 Type Word
22	2	PRCNT	CCA1L	R/W	Channel C Alarm #1 Limit
22	3	PRCNT	CCA2L	R/W	Channel C Alarm #2 Limit
22	6	Status	CCA1T	R	Channel C Alarm #1 Type Word
22	7	Status	CCA2T	R	Channel C Alarm #2 Type Word

Channel	Parameter	Param Type	Mnemonic	Read/Write	Description
25	2	PRCNT	CDA1L	R/W	Channel D Alarm #1 Limit
25	3	PRCNT	CDA2L	R/W	Channel D Alarm #2 Limit
25	6	Status	CDA1T	R	Channel D Alarm #1 Type Word
25	7	Status	CDA2T	R	Channel D Alarm #2 Type Word
31	2	C16	SO70	R	Source for FB98 Output #70
32	2	C16	SO71	R	Source for FB98 Output #71
33	2	C16	SO72	R	Source for FB98 Output #72
34	2	C16	SO73	R	Source for FB98 Output #73
35	2	C16	SO74	R	Source for FB98 Output #74
36	2	C16	SO75	R	Source for FB98 Output #75
37	2	C16	SO76	R	Source for FB98 Output #76
38	2	C16	SO77	R	Source for FB98 Output #77

- i For R - RANGE line in Quick Builder should be set to RANGE 0 38.4
- ii For B - RANGE line in Quick Builder should be set to RANGE -100 100
- iii For RT1 - RANGE line in Quick Builder should be set to RANGE 0 3840
- iv For RT2 - RANGE line in Quick Builder should be set to RANGE 0 3840

Interpreting Status words



Note

Parameter types that are not Status words are described in detail in the *Moore Model 352 SLDC User's Manual*.

Status Word	Go to:
Status Word 02 - Loop 1 Status	page 29

Status Word	Go to:
Status Word 04 - Station Status	page 30
Status Word 05 - Function Block Status	page 30
Status Word 06 - Loop 1 Alarm Status	page 31
Status Word 07 - Channel A Alarm Status	page 31
Status Word 08 - Channel A Status	page 32
Status Word 09 - Channel B Alarm Status	page 33
Status Word 10 - Channel B Status	page 33
Status Word 11 - Channel C Alarm Status	page 34
Status Word 12 - Channel C Status	page 34
Status Word 13 - Channel D Alarm Status	page 35
Status Word 14 - Channel D Status	page 36
Status Word 15 - Loop 2 Status	page 36
Status Alarm Type Words - for Loop1, Channel A, B, C, D	page 37

Status Word 02—Loop 1 Status

bit	meaning
0	automatic (1) / manual (0)
1	local control (1) / non local (0)
2	standby sync (1) / no standby sync (0)
3	external (1) / internal (0)
4	console control (1) / not console (0)
5	computer control (1) / not computer (0)
6	ramping setpoint (1) / no ramping (0)
7	override (1) / no override (0)
8	emergency manual (1) / not em manual (0)
9	configuration hold (1) / not cfg hold (0)
10	HI SP limit (1) / no limit (0)
11	LO SP limit (1) / no limit (0)
12	out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 04—Station Status

bit	meaning
0	not used
1	not used
2	not used
3	database valid (1) / invalid (0)
4	not used
5	not used
6	configuration hold (1) / not cfg hold (0)
7	run mode (1) / config hold (0)
8	not used
9	not used
10	not used
11	config change #1 (1) / reset (0)
12	config change #2 (1) / reset (0)
13	config change #3 (1) / reset (0)
14	error (1) / no error (0)
15	not used

Status Word 05—Function Block Status

bit	meaning
0	alarm #1 FB12 (1) / no alarm (0)
1	alarm #2 FB12 (1) / no alarm (0)
2	deviation alarm FB12 (1) / no alarm (0)
3	flashing bargraph (1) / not flashing (0)
4	lo limit #1 FB09 (1) / no limit (0)
5	hi limit #1 FB09 (1) / no limit (0)
6	lo limit #2 FB51 (1) / no limit (0)
7	hi limit #2 FB51 (1) / no limit (0)
8	emergency local FB98 (1) / not em loc (0)
9	non-updating (1) / normal (0)
10	user status #1 FB15 high (1) / low (0)
11	user status #2 FB15 high (1) / low (0)

bit	meaning
12	emergency internal FB11 (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 06—Loop 1 Alarm Status

bit	meaning
0	alarm #1 (1) / no alarm (0)
1	not acked (1) / acked (0)
2	alarm #1 enabled (1) / not enabled (0)
3	alarm #2 (1) / no alarm (0)
4	not acked (1) / acked (0)
5	alarm #2 enabled (1) / not enabled (0)
6	alarm #3 (1) / no alarm (0)
7	not acked (1) / acked (0)
8	alarm #3 enabled (1) / not enabled (0)
9	alarm #4 (1) / no alarm (0)
10	not acked (1) / acked (0)
11	alarm #4 enabled (1) / not enabled (0)
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 07—Channel A Alarm Status

bit	meaning
0	alarm #1 (1) / no alarm (0)
1	not acked (1) / acked (0)
2	alarm #1 enabled (1) / not enabled (0)
3	alarm #2 (1) / no alarm (0)
4	not acked (1) / acked (0)
5	alarm #2 enabled (1) / not enabled (0)

bit	meaning
6	not used
7	not used
8	not used
9	not used
10	not used
11	not used
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 08— Channel A Status

bit	meaning
0	not used
1	local control (1) / non local (0)
2	not used
3	not used
4	console control (1) / not console (0)
5	computer control (1) / not computer (0)
6	not used
7	not used
8	not used
9	not used
10	not used
11	not used
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 09—Channel B Alarm Status

bit	meaning
0	not used
1	local control (1) / non local (0)
2	not used
3	not used
4	console control (1) / not console (0)
5	computer control (1) / not computer (0)
6	not used
7	not used
8	not used
9	not used
10	not used
11	not used
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 10—Channel B Status

bit	meaning
0	not used
1	local control (1) / non local (0)
2	not used
3	not used
4	console control (1) / not console (0)
5	computer control (1) / not computer (0)
6	not used
7	not used
8	not used
9	not used
10	not used
11	not used

bit	meaning
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 11—Channel C Alarm Status

bit	meaning
0	alarm #1 (1) / no alarm (0)
1	not acked (1) / acked (0)
2	alarm #1 enabled (1) / not enabled (0)
3	alarm #2 (1) / no alarm (0)
4	not acked (1) / acked (0)
5	alarm #2 enabled (1) / not enabled (0)
6	not used
7	not used
8	not used
9	not used
10	not used
11	not used
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 12—Channel C Status

bit	meaning
0	not used
1	local control (1) / non local (0)
2	not used
3	not used
4	console control (1) / not console (0)
5	computer control (1) / not computer (0)

bit	meaning
6	not used
7	not used
8	not used
9	not used
10	not used
11	not used
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 13—Channel D Alarm Status

bit	meaning
0	alarm #1 (1) / no alarm (0)
1	not acked (1) / acked (0)
2	alarm #1 enabled (1) / not enabled (0)
3	alarm #2 (1) / no alarm (0)
4	not acked (1) / acked (0)
5	alarm #2 enabled (1) / not enabled (0)
6	not used
7	not used
8	not used
9	not used
10	not used
11	not used
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 14—Channel D Status

bit	meaning
0	not used
1	local control (1) / non local (0)
2	not used
3	not used
4	console control (1) / not console (0)
5	computer control (1) / not computer (0)
6	not used
7	not used
8	not used
9	not used
10	not used
11	not used
12	link alarms out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Word 15—Loop 2 Status

bit	meaning
0	auto set to “1”
1	local control (1) / non local (0)
2	standby sync set to “0”
3	external/internal set to “0”
4	console control (1) / not console (0)
5	computer control (1) / not computer (0)
6	ramping setpoint (1) / not ramping (0)
7	override set to “0”
8	emergency manual set to “0”
9	configuration hold (1) / not config hold (0)
10	HI SP limit (1) / no limit (0)
11	LO SP limit (1) / no limit (0)

bit	meaning
12	out of service (1) / normal (0)
13	not used
14	not used
15	not used

Status Alarm Type Words—for Loop1, Channel A, B, C, D

bit	meaning
0	Alarm Type
1	
2	
3	Alarm Deadband
4	
5	Delay Time IN
6	
7	
8	Delay Time OUT
9	
10	
11	Ringback/Ringback not required
12	not used
13	not used
14	not used
15	not used

where Alarm Type is enumerated as:

value	meaning
0	no alarm action required
1	HIGH alarm
2	LOW alarm
3	HIGH DEVIATION alarm
4	LOW DEVIATION alarm
5	ABSOLUTE DEVIATION alarm

value	meaning
6	OUT OF RANGE alarm
7	no alarm action required

where Alarm Deadband is enumerated as:

value	meaning
0	0.1% alarm deadband
1	0.5% alarm deadband
2	1.0% alarm deadband
3	5.0% alarm deadband

where Delay Time IN and Delay Time OUT is enumerated as:

value	meaning
0	0.0 seconds
1	0.4 seconds
2	1.0 seconds
3	2.0 seconds
4	5.0 seconds
5	15.0 seconds
6	30.0 seconds
7	60.0 seconds

Mapping between the server and the Moore 352 for bit 0 of the Loop 1 Status (L1ST) address

For ease of control and observation of the Loop 1 Status on the Moore Mycro 352 controller, some values on the server are mapped to different values on the Moore 352 controller. If either a status point or the MD parameter of a point is built with the address L1ST Bit 0 address (channel 9 parameter 1) on a Moore 352 controller, the following mapping is done:

Table 1 Mapping performed between the server and Moore 352

Server Mode	Server Value	Moore 352 Value
MAN-LSP	0	MANUAL/INTERNAL
AUTO-LSP	1	AUTO/INTERNAL
AUTO-RSP	2	AUTO/EXTERNAL

Table 1 Mapping performed between the server and Moore 352 *(continued)*

Server Mode	Server Value	Moore 352 Value
MAN-RSP	6	MANUAL/EXTERNAL

For example, if the MD on a point is set to MAN-RSP, the status of the Loop 1 mode changes to MANUAL and EXTERNAL. The reverse mapping also applies.

**Note**

Mapping is not performed for Loop2. Mapping is only performed for the Moore Mycro 352 controller type. Other bits of the Status LIST address and all other addresses on the device are read from and written to without any value mapping.

Optimizing scanning performance

The maximum amount of data that can be acquired from an controller is influenced by the rate of sending scan packets to the controller. An understanding of the Moore Mycro scan packets will help you configure points so that optimal data acquisition performance can be achieved by maximizing the amount of data acquired with each scan packet.

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. See the *Configuration Guide* for your server for usage of **lisscn**.

Moore Mycro scan packets

Up to 100 controller addresses using the same scan period and accessing parameter 1 of the Moore Mycro controller can be combined into a single scan packet. However, addresses not accessing parameter 1 but using the same scan period can only be combined in groups of 5 into a single scan packet. Therefore, to minimize the number of scan packets, use slower scan periods for all points not accessing parameter 1 and try to use the same scanning period for all these points as well.

Server and Station tasks for Moore Mycro

4

This chapter describes tasks for the Moore Mycro controller that you perform either on the server or from any Station.

For:	Go to:
Testing communications with the server	page 42
Troubleshooting point configuration errors	page 43

Testing communications with the server

Use the Moore Mycro test utility, **mmutst**, to test communications between the server and the controllers after you have downloaded your channel and controller definitions to the server database.

You can also use **mmutst** to change the values of registers that in turn set configuration items on the Moore PLC.

Before testing:

- 1 Set up and configure controllers.
- 2 Connect all cables.
- 3 Enter definitions for controller and channel in Quick Builder project file.
- 4 Downloaded project file definitions to the server without errors.

To run the test utility from a Command Prompt window:

- 1 Open a Command Prompt window and type **mmutst** then press ENTER.
- 2 When prompted, enter the channel number of your Moore Mycro channel. You are prompted with a menu of the test functions.
- 3 Follow the instructions to test communications between the server and your controller(s).

Troubleshooting point configuration errors

Errors while downloading to host

If points are configured with illegal configuration details this might cause problems when they are downloaded to the server. If this occurs, read the output file created and correct the errors.

Note that certain combinations of server point types and Moore Mycro 352 addresses are illegal. All Status Words on the device (addresses looking at certain bits) must be defined using the STA tag type or referenced from the MD parameter of a point. See “Addresses supported by the interface” on page 23 for the addresses which are Status Words. Also, analog and accumulator points cannot specify a bit field.

Unexpected point values

If a point does not have the expected value, the point might be configured incorrectly. Ensure that the point source address has the correct data type. If not, either check the point source address in the Quick Builder line or the device address definition file **mmu_352_def**.

If the point is a STAND format, check the RANGE specified. If the value conflicts with a value for a loop value on the faceplate of the Moore, check that the RANGE matches the Process Lo, Hi, and Decimal Point which is configured in FB15.

For some values, the point value reported by server will be slightly different from the value shown on the faceplate of the Moore Mycro due to rounding errors on the device.

Index

A

- addressing
 - address format option 1 examples 21
 - address format option 2 examples 22
 - addresses supported 23
 - Loop 1 Bit 0 Status LIST address mapping 38
 - Moore Mycro 352 controller 20
- architecture, sample system designs 8

C

- channel definition
 - for Moore Mycro 352 14
- communications
 - Moore Mycro 352 11
- configuring Moore Mycro 352, steps for 5
- controller
 - addressing Moore Mycro 352 20
 - definition for Moore Mycro 352 18

D

- documentation
 - for Moore Mycro 352 6

G

- generic channels
 - Port tab 15

M

- Moore Mycro 352
 - controller setup 9

P

- parameter data types 22
- Port tab 15

- Property Page
 - Port tab 15

S

- scanning
 - optimizing for Moore Mycro 352 40

OPC Client Interface

Issue

10

Date

November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for OPC Client	6
	OPC Client support	6
	OPC server documentation	6
	OPC-specific Terms.	7
2	OPC Client interface setup	9
	Description of OPC	10
	How OPC data transfer works.	10
	Callback.	10
	Synchronous read request	11
	Architectures	12
	Valid OPC interface configurations.	12
	OPC Client controllers, channels and points.	13
3	OPC Client controller configuration and addressing	15
	Defining an OPC Client channel	16
	OPC Client channel main properties.	16
	Defining an OPC Client controller	19
	OPC Client controller main properties	19
	Defining an OPC Client address for a point parameter value	21
	Entering an address	21
	Optimizing scanning performance	24
	OPC Client scan packets	24
4	Server and Station tasks for OPC Client	25
	Testing communications with the server.	26
	Troubleshooting OPC Client communication errors.	26

Getting started

1

This reference describes how to set up, configure and test the OPC Client Interface. (For a summary of other OPC interfaces, and their respective uses, see the *Configuration Guide*.)

This reference contains detailed information for configuring the OPC Client Interface using Quick Builder.

Steps for connecting and configuring an OPC Client controller.

Complete each step before commencing the next step.

Steps	Go to:
Install and set up the OPC server as specified in the documentation supplied by the OPC server manufacturer.	
Install any OPC server configuration files on the OPC client computer as specified in the documentation supplied by the OPC server manufacturer.	
Test communications between the OPC client computer and the OPC server computer.	page 26
Define channels with Quick Builder.	page 16
Define controllers with Quick Builder.	page 19
Download channel and controller definitions to the server.	
Define points to address OPC server items with Quick Builder.	page 21

Support and documentation for OPC Client

OPC Client support

The OPC Client Interface is an OPC client which supports communications to OPC servers that meet the specification of the OPC Data Access Standard version 1.0a or 2.0.

The OPC Client Interface supports both local and remote OPC servers.

OPC server documentation

The documentation supplied by the OPC server manufacturer is relevant, and should be read before creating any OPC Client controllers.

OPC-specific Terms

CLSID

CLSID is the class identifier that identifies an object. An object registers its CLSID in the system registration database so the object can be loaded and programmed by other applications.

group

A number of items with the same update rate and deadband.

item

A single data source of the OPC server.

OPC

OPC stands for OLE for Process Control. It is a set of standards that define sets of COM interfaces (based on Microsoft's COM/OLE technology) to be observed by OPC clients and servers. This set of standards was established by the OPC Foundation to foster greater interoperability between automation and control applications, field systems and devices, and business and office applications. When the term OPC is used in this Reference, it refers specifically to the OPC Data Access Standard.

ProgID

A programmatic identifier. A registry entry that can be associated with a class identifier (CLSID). The format of a ProgID is

<vendor>.<component>.<version>, separated by periods and with no spaces.

The ProgID identifies a class, but with less precision. The ProgID is used to identify the OPC server from other COM/DCOM components on the same computer. For more details, see any setup and installation instructions provided by the OPC server manufacturer.

update rate

The internal update rate of the items in the OPC server.

OPC Client interface setup

2

This chapter describes how to set up an OPC Client Interface.

For details about:	Go to:
Description of OPC and how data transfer works	page 10
Supported architectures	page 12
How Quick Builder uses controllers, channels and points to configure the OPC Client Interface	page 13

Description of OPC

OPC provides data from a data source and communicates the data to any client application in a standard way, thereby eliminating the requirement for an application to have specific knowledge about a particular data source, such as internal structure and communications protocols.

An OPC server and an OPC client, can reside either on the same computer (local server) or on different computers (remote server).

How OPC data transfer works

OPC server data is available to the OPC client as *items*. To receive items from the OPC server, the OPC client must gather one or more items into a *group*.

The OPC client requests the OPC server to create a group with a client-specified maximum (at most) update rate and a deadband. The OPC client then requests the OPC server to add items to the group. The update rate and the deadband of a group apply to all items in that group.



Note

- There is no concept of hardware in the OPC Data Access Standard; there are just items. An OPC server may represent a piece of hardware as an item and the item's value may indicate the state of the hardware. However, whether or not such representation is available is server-specific.
 - Although the OPC client can specify any update rate for a group, the OPC server decides whether the request is honored.
-

Callback

Generally, the OPC server sends data to OPC clients through callbacks. After a group has been created, the OPC server creates a cache for the group items. The cache is updated according to the group's update rate. The OPC server sends only updated values to the OPC client for items in the group if there has been significant change since the last cache update (based on the group's deadband).

This method of data update significantly reduces traffic between the OPC client and the OPC server as there is no need for periodic read requests to the OPC server. An OPC client gets data when there is a significant change. The level of change required to trigger an update from the OPC server is defined by the OPC client.

Synchronous read request

The OPC client can also send synchronous read requests to the OPC server, independent of the server callbacks. The OPC client specifies whether the data should come from the OPC server's internal cache or from the field/hardware device. This method of scanning is less efficient than callback.

Architectures

For the OPC Client Interface to communicate with an OPC server, the OPC server must be installed on an appropriate computer. (The OPC Client Interface and the OPC server can reside on the same computer or on different computers.)

To install an OPC server, install and set up the OPC server as specified in the documentation supplied by the OPC server manufacturer.

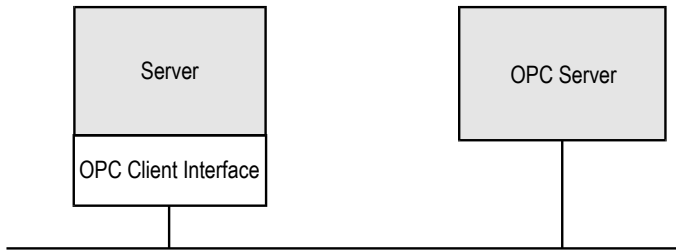
Valid OPC interface configurations

The server OPC interface supports single-channel communications.

Single channel communications

A single-channel configuration.

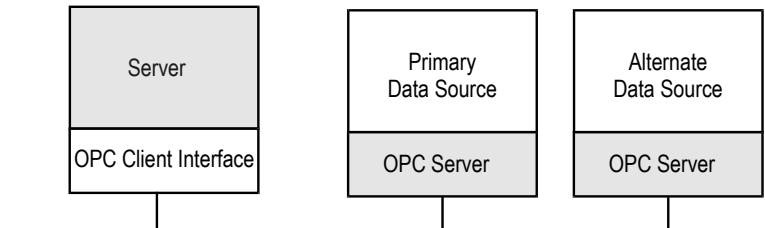
Figure 1 OPC Interface Single-channel Configuration



Alternate data source

The OPC Interface also supports an alternate data source. If the OPC Interface connection to the primary data source fails, the OPC Interface connects to the alternate data source.

Figure 2 OPC Redundant Alternate Data Source Configuration



OPC Client controllers, channels and points

An OPC Client controller is Quick Builder's mechanism for configuring one or more groups with the same deadband. Unlike most other types of controller, an OPC Client controller is an abstract entity, and does not represent a physical device.

Similarly, a point parameter is Quick Builder's mechanism for defining an item on an OPC Client controller. During configuration, you map each item to an appropriate point parameter on an OPC Client controller. Only floating point number data types are supported.

An OPC Client controller manages groups as follows:

- A group is created for each scan period used by the point parameters (items) defined on the controller. (That is, all point parameters that have the same scan period are placed in the same group.) Note that if a scan period of 0 is used, the item is placed in a group with an update rate equal to the slowest scan rate of the server system.
- All points defined on the controller have the same OPC deadband. (Note that the OPC deadband is not the same as the alarm and control deadbands that can be specified for analog points.)
- The maximum number of items that can be configured in a controller is 963.

An OPC Client channel forms the interface between one or more OPC Client controllers and an OPC server. You configure an OPC Client channel in the same way as other channels, however, two properties have special significance as far as OPC is concerned:

- **Diagnostic Scan Period.** This determines how often the OPC client tests the status of the OPC server to ensure that it is OK.
- **Background Scan Period.** This applies to all controllers configured under the channel and determines how often the OPC client explicitly reads all items configured on a controller if that controller is configured with background scanning enabled. This is independent of the callbacks.

OPC Client controller configuration and addressing

3

This chapter describes how to configure OPC Client controllers using Quick Builder.



Note

An OPC Client controller is an abstract entity that is used as a container for OPC groups. An OPC Client controller does not represent a real controller or a real piece of hardware.

For:	Go to:
Steps for defining an OPC Client channel	page 16
Steps for defining an OPC Client controller	page 19
Defining an address for a point parameter value	page 21
How to optimize scanning performance	page 24

Defining an OPC Client channel

For an introduction to OPC Client channels and controllers, see “OPC Client controllers, channels and points” on page 13.

To define a channel using Quick Builder:

- 1 Click  to open the Add Items dialog box.
- 2 Select channel from **Add Items**.
- 3 Select OPC Client from **Type**.
- 4 Specify the Main tab property values. See “OPC Client channel main properties” on page 16.

OPC Client channel main properties

The Main tab defines the properties of an OPC Client channel.



Note

If you specify an alternate Host Name, Quick Builder builds a redundant OPC channel. See the *Configuration Guide* for your server for information about monitoring redundant channels.

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description	(Optional) A description of the channel. A maximum of 30 characters, including spaces.

Property	Description
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. 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 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for 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 $[\sqrt{9} \times 25 = 75]$.
Fail Alarm Limit	The communications alarm fail limit at which the channel is declared to have failed. Set this to double the value specified for the channel Marginal Alarm Limit .
Connect Timeout	Amount of time, in seconds, the server waits to connect to the OPC server before abandoning the connection. The default is 10 seconds.
Read Timeout	Amount of time, in seconds, the server waits for a reply from the OPC server after a synchronous read request. The default is 2 seconds.
Host Name (preferred)	The name of the computer on which the preferred OPC server software resides. If the OPC server is on the same computer as the OPC Client Interface, the name must be <code>LocalHost</code> .
Host Name (alternate)	(Optional) The name of the computer on which the alternate OPC server software resides. If it is on the same computer as the OPC Client Interface, the name must be <code>LocalHost</code>. Note: If you specify an alternate host name, Quick Builder builds a redundant channel OPC connection. See the <i>Configuration Guide</i> for the server for information on monitoring the status of redundant channels.
Diagnostic Scan period	The amount of time, in seconds, between diagnostic scans. The diagnostic rate must be set to one of the valid server scan periods. The default is 60 seconds. This value is used as the rate for sending synchronous requests for checking the OPC server's current status.

Property	Description
Background Scan Period	The background scanning period (in seconds). This must be set to one of the valid server scan periods (defaults to 60). This value is used as the rate for sending explicit read requests of all items configured under a controller if that controller is also configured with background scanning enabled.
ProgID	The ProgID for the OPC server that is to be connected.
OPC server host time can drift	If selected, indicates that the OPC server computer's clock (UTC time) is known to drift. This can occur if the clock is synchronized to an external clock. Moving in or out of daylight savings time is not a time drift as the UTC time does not change.
Item Type	Shows the channel's type.
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	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

Defining an OPC Client controller

For an introduction to OPC Client channels and controllers, see “OPC Client controllers, channels and points” on page 13.

To define a controller using Quick Builder:

- 1 Click  to open the Add Items dialog box.
- 2 Select Controller from **Add Items**.
- 3 Select OPC Client from **Type**.
- 4 Specify the Main tab property values. See OPC Client controller main properties.

OPC Client controller main properties

The Main tab defines the properties for an OPC Client controller.

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description	(Optional) A meaningful description for this controller. A maximum of 30 characters, including spaces.
Channel Name	The name of the channel on which the controller communicates. You need to have defined the channel in order for it's name to appear in the list.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 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 value is reached, an urgent alarm is generated. Set this to double the value specified for the controller Marginal Alarm Limit .

Property	Description
Background Scan	Specify whether background scanning is performed in addition to normal callback operation. Select Enabled if you do not want the default (Disabled). The background scan period is configured under the channel.
Deadband	The OPC deadband, which applies to all items referenced by point parameters belonging to this controller. The deadband Indices are defined as: 0 = 0.000% 1 = 0.001% 2 = 0.002% 3 = 0.005% 4 = 0.010% 5 = 0.020% 6 = 0.050% 7 = 0.100% 8 = 0.200% 9 = 0.500% 10 = 1.000% 11 = 2.000% 12 = 5.000% 13 = 10.000% 14 = 20.000% 15 = 50.000%
Item Type	Shows the controller's type.
Last Modified	Shows the date of the most recent modification to this controller's property details.
Last Downloaded	The date that the item was last downloaded to the server.
Item Number	The unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field 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.

Defining an OPC Client address for a point parameter value

A point parameter is Quick Builder’s mechanism for “mapping” a single item on an OPC Client controller. When configuring a point parameter, note that:

- The point parameter’s **Scan Period** is used as the item’s update rate.
- Periodic scanning is based on OPC callbacks, not on OPC synchronous read requests.

Entering an address

For **Source Address** and **Destination Address** the format for a OPC Client controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the OPC Client controller.
<i>Location</i>	The location in the controller where the value is recorded. See “Location syntax” on page 22.

If you would like help when defining an address, click  next to **Address** to display Address Builder. For details, see the help.

Location syntax

The format for the location is:

[AccessPath] OPCItemName [DataFormat]

Part	Description
<i>[AccessPath]</i>	The access path is not normally required. However, it is required for OPC servers developed using the Rockwell OPC Toolkit. You need to see the vendor's documentation to determine whether an access path is required. The access path generally represents a device driver or topic (if the server was previously developed as a DDE server). Note, if you do specify an access path: - You must include the square brackets. (In this particular case, they do not simply indicate an optional part of the syntax.) - There is no space between the access path and the <i>OPCItemName</i>.
<i>OPCItemName</i>	The location in the controller where the value is recorded. You need to see the vendor's documentation for the syntax. For example, if you are using the system OPC server to access the PV of a point named <i>sinewave</i> the location would be <i>sinewave.PV</i>. A maximum of 64 characters. Any printable ASCII character that you can type, including the space character, is supported.
<i>DataFormat</i>	The data format. If you do not specify a data format, the default is IEEEFP. See "Data format definitions" on page 22.

Data format definitions

If you want to use a user-defined data format, you must define the format on the server. For details, see the *Configuration Guide*.

Format	Counts
IEEEFP	IEEE single-precision floating point
U3BCD	0 to 999 BCD
U4BCD	0 to 9999 BCD
U4095	0 to 4095

Format	Counts
U999	0 to 999
U9999	0 to 9999
U100	0 to 100
U1023	0 to 1023
U8B	0 to 255
S16B	-32768 to 32767
S8B	-128 to 127
S9999	-9999 to 9999
U16B	0 to 65535
U15B	0 to 32767
U14B	0 to 16383

Optimizing scanning performance

The maximum amount of data that can be acquired from an controller is influenced by the rate of sending scan packets to the controller. An understanding of the OPC scan packets will help you configure points so that optimal data acquisition performance can be achieved by maximizing the amount of data acquired with each scan packet.

The scan packets that have been built can be listed by using the **lisscn** (list scan) utility. Listing scan packets helps verify the scanning strategy. See the *Configuration Guide* for your server for usage of **lisscn**.

OPC Client scan packets

OPC groups are collections of items with the same callback period and deadband. In order to reduce the number of OPC groups (and hence OPC scan packets) you should:

- Assign all points with the same deadband to the same controller, remembering a maximum of 963 items can be assigned to a single controller.
- Reduce the number of different scan periods used for points on a controller.

Server and Station tasks for OPC Client

4

This chapter describes tasks for the OPC Client controller that you perform either on the server or from any Station.

For:	Go to:
Testing communications with the server	page 26
Troubleshooting scanning errors	page 26

Testing communications with the server

You use the OPC Client test utility, **opctst**, to test the:

- Connection to an OPC server
- Creation of OPC groups on a server
- Addition of OPC items into groups
- Reading and writing of values to OPC items on the OPC server computer

Before you use **opctst** you need to:

- Complete any OPC server setup and configuration as required. See the setup and installation instructions supplied by the OPC server manufacturer.
- Install any OPC server configuration files on the OPC client computer as required. See the setup and installation instructions supplied by the OPC server manufacturer.
- Ensure you are logged on using the **mng**r account as this is the account that the OPC Client Interface uses to communicate with the OPC server.

To run the opctst utility:

- 1 Open a Command Prompt window, type **opctst** and press ENTER.
- 2 When the Windows application starts, select the OPC menu option and perform each of the menu operations in turn (that is, Initialize COM, Connect to OPC server, and so on).

Troubleshooting OPC Client communication errors

If you experience difficulty getting the OPC Client Interface to communicate with an OPC server, refer to the system server log file (accessible via an icon in the Diagnostic Tools folder). The log file gives an indication as to the cause of any OPC communications problems.

Index

A

alternate data source 12
architectures 12

C

channel, defining 16
communication errors 26
configurations 12
configuring OPC, steps for 5
controller, defining 19

O

OPC

callback, described 10
data transfer, described 10
group, described 7
item, described 7
overview 10
synchronous read request, described 11
update rate described 7

OPC Client Interface

architectures 12
communication errors 26
configurations 12
controllers and channels, using 13
defining 19
defining a channel 16
optimizing scanning performance 24
point parameter address 21
points, using 13
scan packets 24
support and documentation 6
testing communications 26

P

point parameter address, defining 21

S

scanning, optimizing performance 24
single channel communications 12

T

testing communications 26

Siemens S5 and S7 Interface Reference

Issue

8

Date

November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for Siemens	6
	Device support	6
	Other documentation on Siemens	6
	Siemens-specific terms	7
2	Siemens setup	9
	Architectures for Siemens	10
	Setting up a Siemens controller	11
	Configuring an S7 controller	11
	Configuring an S5 controller	11
	Setting up the server	12
	Multiple CP 1613 cards in the server	12
	Reserving memory for dual-port RAM	12
	Memory hole	12
	Configuring the communications processor database (S7)	12
	Configuring the communications processor database (S5)	14
	Finalizing the database	15
3	Siemens configuration and addressing	17
	Defining a Siemens channel	18
	Siemens channel main properties	18
	Defining a Siemens controller	20
	Siemens controller main properties	20
	Defining a Siemens address for a point parameter	22
	Entering an address	22
	Location syntax (S5)	22
	Location syntax (S7)	24
	Optimizing scanning performance	27
4	Server tasks for Siemens	29
	Testing Siemens communications	30
	Testing Siemens S5 controllers	30
	Testing Siemens S7 controllers	30
	Troubleshooting point configuration errors	32
	Troubleshooting S5 controllers	32
	Troubleshooting point configuration errors on S7 controllers	32

CONTENTS

Diagnostic failures on S7 controllers 33

Getting started

1

This reference provides the information you need to set up, configure, and test Siemens S5 and S7 Series controllers.

Steps for connecting and configuring a Siemens.

Complete each step before commencing the next step.

Step:	Go to:
Set up the controller and network	page 11
Set up the server	page 12
Define channels using Quick Builder	page 18
Define controllers using Quick Builder	page 20
Download channel and controller definitions to the server	
Test communications	page 30
Define controller points using Quick Builder	page 22

Support and documentation for Siemens

Device support

The server supports Siemens S5 and S7 series controllers, as listed in the following table. Note how the communications processor (CP) card required varies according to the controller model.

Controller Model	Communications Processor Card
S5-115U	CP 1430 TF
S5-135U	CP 1430 TF
S5-155U	CP 1430 TF
TI-505 series	CP 1434 TF
S7-300	CP 343-1
S7-400	CP 443-1

The server requires a CP 1613 card.

The server supports the following versions of Siemens CP 1613 drivers:

- Please consult the Software Change Notice or Software Release Bulletin.



Note

- Not all memory addresses exist on all controller types. See the Siemens documentation for more detail on what addresses are available for each type.
 - The Siemens S5 interface for Windows NT does not permit operation over the Siemens SINEC TF on an L2 network.
-

Other documentation on Siemens

For detailed information concerning the configuration of each type of Siemens controller see the Siemens SIMATIC NET Electronics Manuals CD-ROM.

Siemens-specific terms

Application association

A logical link between the communications processor (CP) card in the server and the CP card in a controller.

CP card

Communications processor card. Each Siemens controller, as well as the server, requires a CP card.

VFD

Virtual Field Device. A VFD is a logical link between the CP card in the server and the CP card in the controller. The server uses the VFD to address the controller, and to acquire and control its data.

Siemens setup

2

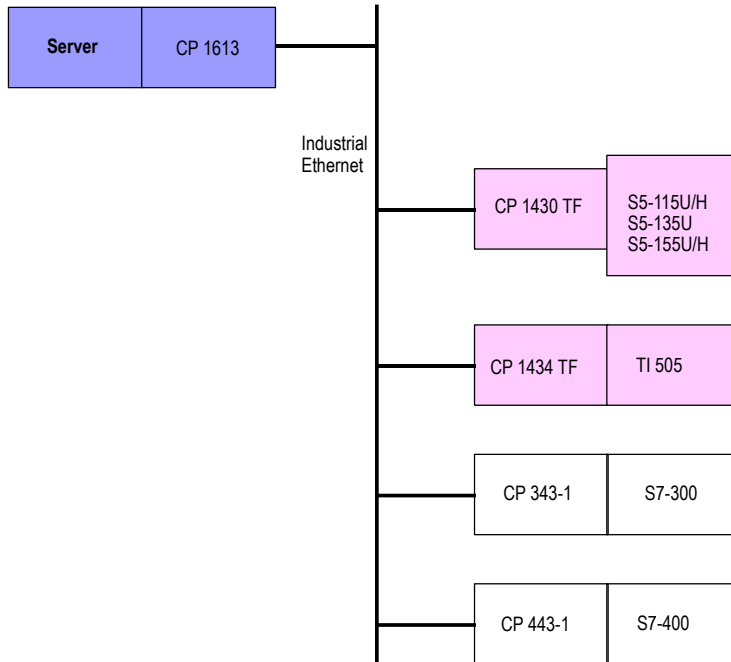
This chapter describes how to set up a Siemens controller.

The tasks for setting up a Siemens controller are:

For:	Go to:
Supported architectures	page 10
Setting up a controller	page 11
Setting up the server	page 12

Architectures for Siemens

The following figure shows the basic architecture.



Note

- The Siemens S5 and S7 interfaces do not support multiple CP1613 communication processor cards in the server.
 - The CP1613 communication processor card in the server can only be used for S5 or S7 communication, not both. If you have both S5 and S7 series controllers, you should use the OPC interface instead of the S5 or S7 interface.
 - You can build multiple controllers on a single controller, each using their own connection in the Siemens software. However, you cannot configure redundant links to Siemens controllers.
-

Setting up a Siemens controller

You must install the appropriate communications module in each Siemens controller and assign a MAC (hardware) address for each card. The controllers communicate with the server via an Ethernet LAN compliant with IEEE 802.3.

Configuring an S7 controller

When working with an S7 controller, use the Siemens Step7 software with the NCM S7 optional package installed to assign the Ethernet address.

See the Siemens documentation for more information.

Configuring an S5 controller

When working with an S5 controller, use the Siemens Step5 software with the COM 1430TF optional package installed to assign the Ethernet address.

See the Siemens documentation for more information.

Application associations (S5 Only)

An *application association* is a logical link between the CP card in the server and the CP card in the controller. The application association(s) must be compatible in both communications processors. The server uses the application associations to address individual CP cards and so acquire and control the data in the controller registers. You can define multiple application associations between the server and a controller.

Consult the Siemens documentation for details on configuring application associations in the controller.

Variables (S5 Only)

The server communicates with the CP card in the controller using Siemens' Variable Access Services. Internal registers in the controller are not accessible by the Siemens interface. Instead, map a variable into the internal registers of the controller, which is then accessed by the server. The variables must be defined using loader software and downloaded into the CP card.

Consult the Siemens documentation for details on configuring variables in the controller.

Setting up the server

You need to install and configure the CP 1613 card in the server. See the appropriate Siemens documentation for details.

Multiple CP 1613 cards in the server

The server can only use one CP 1613 card. This card can support either S5 or S7 communications. The Siemens S5 and the Siemens S7 interfaces can both have multiple channels using one card.

Reserving memory for dual-port RAM

The Siemens manual refers to using a free memory area as dual-port RAM to avoid memory conflicts—however, this is not possible on DELL computers.

When adding any plug-in cards, check that there are no memory conflicts.

Memory hole

The Siemens manual refers to enabling a memory hole—do not do this.

Configuring the communications processor database (S7)

Generate the communications processor database with the COML S7 program.

Figure 1 Typical database settings for an S7 controller

The screenshot shows the SIMATIC NET COM1 S7 -SERVER_A.TXT dialog box. The 'Node name' is 'Server A' and the 'Network type' is 'ETHERNET'. The 'S7 connection list' on the left contains 'Connection1'. The 'Edit S7 connection selected on left' panel on the right shows the following settings: Name: 'Connection1', VFD Name: 'VirtualFieldDev1', Remote Addr: '08.00.06.01.00.02', Local TSAP: '01.00' (hexadecimal), Remote TSAP: '01.02' (hexadecimal), and Device ID: PG 1 Rack 0 Slot 2. Buttons for 'Delete', 'Insert', 'Replace', and 'Cancel' are visible at the bottom.

Property	Description
S7 connection list	A Connection is a representation of a single link to a configured VFD. The server architecture supports only one connection per VFD.
VFD Name	A Virtual Field Device (VFD) is a logical link between the CP card in the server and the CP card in the controller. The server uses the VFD to address the controller, and to acquire and control its data. 18 characters maximum. Typically, you configure one VFD for each controller on the network. However, you can define multiple VFDs for a controller to segregate the points built on a single controller into logical groups. This does not affect the bandwidth requirements of point scanning. You can configure a maximum of 3 VFDs per physical controller.
Remote Addr	The Remote Address is the address of the CP card in the controller, as configured in the Siemens software.
Local TSAP	Always configure the Local TSAP as 01.00.

Property	Description
Remote TSAP	Configure the Remote TSAP for the rack and slot number of the CPU module in the physical device. The server communicates with this CPU using the configured VFD and Connection names.

Symbolic Addresses

The server communicates with the controller using Siemens symbolic addresses. These addresses specify the location and type of data to be acquired or controlled in the controller. See “Defining a Siemens address for a point parameter” on page 22.

Configuring the communications processor database (S5)

Generate the communications processor database with the **COML TF** program.

The database associates the parameters of the CP card in the server with those in the controller. Specifically, the **Application Association Name**, **Remote Address**, **Remote TSAP**, **Connection Establishment** and **Connection Type** must match the controller settings. Similarly, the communications processor in the controller must be configured with a remote TSAP the same as the server’s **Local TSAP**.

Figure 2 Typical database settings for an S5 controller

The screenshot shows the 'COML TF -SERVER_A.TXT' dialog box in the SIMATIC NET environment. The 'Node name' is 'Server_A', 'Station address' is '08.00.06.01.00.01', and 'CP type' is 'CP 1613 TF'. The 'Application association list' on the left contains 'Server_A'. The 'Edit application association selected on left' section shows the following settings: Name: 'Server_A', Remote Addr: '08.00.06.01.00.03', Local TSAP: 'PC 1' (hex: '50.43.20.31'), Remote TSAP: 'PLC 1' (hex: '50.4C.43.20.31'), and Server-ID: 'server'. Under 'Connection establ.', 'active' is selected. Under 'Connection type', 'static' is selected. The 'PDU size' is set to '1000' and 'TPDU size' is '1024'. Buttons for 'Delete', 'Insert', 'Replace', and 'Cancel' are at the bottom.

Finalizing the database

Once you have generated the database, save it in both text and binary forms. The text form gives you a base for future modifications, while the binary format is loaded by the CP card on system startup.

Siemens configuration and addressing

3

This chapter describes how to configure a Siemens controller using Quick Builder.

For information about:	Go to:
Defining a Siemens channel	page 18
Defining a Siemens controller	page 20
Defining an address for a point parameter value	page 22
Optimizing scanning performance	page 27

Defining a Siemens channel

To define a channel:

- 1 Click  to display the Add Items dialog.
- 2 Select channel as the item and the appropriate Siemens type (SiemensS5 or SiemensS7).
- 3 Set the property values on the Main tab—see “Siemens channel main properties” on page 18.

Siemens channel main properties

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

Property	Description
Name	The unique name of the channel. A maximum of 10 alphanumeric characters (no spaces or double quotes).
Description	(Optional) A description of the channel. A maximum of 30 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. 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 2 or more, depending on the error and decrements for each good call. To calculate an acceptable limit, multiply the square root of the number of controllers on the channel by the Marginal Alarm Limit defined for 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 $[3 \text{ is square root }] \times 25 = 75$.
Fail Alarm Limit	The communications alarm 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 in Marginal Alarm Limit.
Diagnostic Scan Period	The period, in seconds, between diagnostic scans which verify communications integrity with the controller. The default is 60 seconds.

Property	Description
Device Name	Only applicable to S7 channels. The device name of the Siemens CP 1613 card. Normally, CP_H1_1:.
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. 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.

Defining a Siemens controller

To define a controller:

- 1 Click  to display the Add Items dialog.
- 2 Select controller as the item and the appropriate Siemens type (SiemensS5 or SiemensS7).
- 3 Set the property values on the Main tab. See “Siemens controller main properties” on page 20.

Siemens controller main properties

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

Property	Description
Name	The unique name of the controller. A maximum of 10 alphanumeric characters (no spaces or double quotes). For LAN connected controllers, the name must not contain underscore (_) characters. This name is used to look up the IP address in the TCP/IP database if you do not specify an IP Address property.
Description	(Optional) A description of the controller. A maximum of 30 alphanumeric characters, including spaces.
Channel Name	The Siemens channel on which the controller communicates with the server.
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 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. Set this to double the value specified in Marginal Alarm Limit .
Controller Type	(Only applicable to S5 controllers.) The model of the physical controller.

Property	Description
Link	(Only applicable to S5 controllers.) The logical link or application association between the communications processor modules in the server and in the controller. 18 characters maximum. This entry allows connection to more than one CP module.
Alternate Link	(Only applicable to S5 controllers.) The name of the alternative application association. This link is required by Batch Supervisor only. 18 characters maximum.
Timeout	(Only applicable to S5 controllers.) The time, in seconds, that can elapse before a communication request is deemed to have failed. The default is 5 seconds. Note that if you make the timeout too brief, the Siemens interface on the server may stop. You can test the timeout setting using the test utility, smntst . See “Testing Siemens communications” on page 30.
Connection	(Only applicable to S7 controllers.) The name of the connection that the server uses to communicate with the controller.
VFD Name	(Only applicable to S7 controllers.) The name of the VFD that the server uses to communicate with the controller.
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 controller’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. 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.

Defining a Siemens address for a point parameter

Entering an address

For **PV Source Address**, **Source Address**, and **Destination Address** the format for an Siemens controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the Siemens controller.
<i>Location</i>	The address within the controller where the value is stored. See: • “Location syntax (S5)” on page 22 • “Location syntax (S7)” on page 24

If you would like help when defining an address, click  next to **Address** to display Address Builder. For details, see the help.

Location syntax (S5)

The format for the location is:

Variable Offset [DataFormat|BitNumber]

Part	Description
<i>Variable</i>	The name of the variable configured in the controller’s CP card. It can be a simple variable or an array.
<i>Offset</i>	The element index in the variable. For a simple variable, the value is 0.
<i>DataFormat</i>	Only applicable to Analog point parameters. See “Data formats (S5)” on page 23. The default format is U16B.

Part	Description
<i>BitNumber</i>	The bit number is used to specify bits in the register. It corresponds to the starting bit, and valid bit offsets range from 0 to 15, with bit 0 being the most right-hand bit. The point type determines how many bits are used for the point parameters. For status points, it is implicit and determined by the width of the parameter. (For a 3-bit status point, the PV uses three consecutive bits starting at the bit number, for example, if the bit number is 7; bits 7, 8 and 9 are used). For analog and accumulator points, all bits starting from the bit number are used (for example, if the bit number is 10; bits 10, 11, 12, 13, 14, and 15 are used). In this case the value is not scaled by the server range. The default value for the bit number is 0.

Example

A PV source address for the controller SMNANA1

SMNANA1 VAR1 0 U9999

Data formats (S5)

If you want to use a user-defined data format, you must define the format on the server. See the *Configuration Guide* for details.

Data Format	Description	Scaled by Server Range
S8B	-128 to 127	Yes
S16B	-32768 to 32767	Yes
S32B	-2147483648 to 2147483647	No
U8B	0 to 255	Yes
U15B	0 to 32767	Yes
U16B	0 to 65535	Yes
U9999	0 to 9999	Yes
IEEEFP	Single-precision floating point	No

Location syntax (S7)

The format for the location is:

Variable [*B:BitNumber*|*DataFormat*]

Part	Description
<i>Variable</i>	The name of the variable configured in the controller's CP card.
<i>BitNumber</i>	(Only applicable to status points.) The valid range is 0 (default) to 15, where 0 is the right most bit in the register. The number of bits used for the point parameter is implicit and is determined by the width of the parameter.
<i>DataFormat</i>	(Only applicable to Analog point parameters.) Explicit data formatting is not required when defining S7 variable names because the name implicitly contains the data format. See "Supported symbolic addresses (S7)" on page 24. You can override automatic data formatting by explicitly specifying the format, but this practice is not recommended. Only override a data formats as a last resort and check that the server retrieves the desired format before forcing the data format for any other point parameters. See "Data formats (S7)" on page 26.

Supported symbolic addresses (S7)

The server utilizes the Siemens SAPI-S7 interface for device communications. Consequently, the server accesses S7 addresses in accordance with valid S7 symbolic addresses. See the Siemens document, *S7 Programming Interface*, for details.

The following table lists example symbolic address names supported by the server, and indicates whether you can override the default data format. The examples are only valid if the S7 controller has been configured with appropriate data areas and/or peripherals.

Address Examples	Default Data Format	Point Type(s)	Data Format Override Possible
DB2,INT4 DI2,INT4	S16B	Analog	Yes
DB1,DINT8 DI1,DINT8	S32B	Analog	Yes

Address Examples	Default Data Format	Point Type(s)	Data Format Override Possible
DB1,B1 DB2,BYTE2 DI1,B1 DI2,BYTE2	U8B	Analog	Yes
DB1,W10 DB1, WORD1 DI1,W10 DI1, WORD1	U16B or bit position	Status / Analog	Yes (Analog)
DB1,D4 DB1,DWORD4 DI1,D4 DI1,DWORD4	U32B	Analog	Yes
DB2,REAL12 DI2,REAL12	IEEEFP	Analog	Yes
DB1,X14.2 DI1,X14.2	Bit position	Status	No
Z1	U32B	Accumulator	No
C1	U32B	Accumulator	No
T5	U32B	Accumulator	No
A5	U32B or bit position	Status (Source Only) / Analog	No
E5	U32B or bit position	Status (Source Only) / Analog	No
MB1,1	U8B	Analog	Yes
MB1,2	U16B	Analog	Yes
MW2,2	U32B	Analog	Yes
MD4,1	U32B	Analog	Yes

**Note**

- The number specifying the location in the data block (for example, **DB1**, **B1**, **DB2**, **REAL5**, **MW2**, **1**) is always the byte offset into the data block. In the case of **DB1**, **DINT4**, the number 4 is the byte offset into data block 1 and not a double word offset.
- Address declarations can overlap. For example, **DB1**, **W0** and **DB1**, **B1** share the same lower byte and **DB1**, **DWORD0** contains both **DB1**, **W0** and **DB1**, **B1**.
- When using the address format **DB1**, **X14.2**, no bit position can be specified since the last digit (2 in this example) is the bit position. All other status points must specify the starting bit position by using the syntax **B : x** where *x* is the bit position. The following example shows how to specify reading from bit 1 of the S7 symbolic address **DB1**, **WORD3**.

Example

A PV source address for the controller SMNSTA1

SMNSTA1 DB1,WORD3 B:1

Data formats (S7)

User-defined data formats are not supported for the Siemens S7 interface.

Data Format	Description	Scaled by Server Range
S8B	-128 to 127	Yes
S16B	-32768 to 32767	Yes
S32B	-2147483648 to 2147483647	No
U8B	0 to 255	Yes
U16B	0 to 65535	Yes
U32B	0 to 4294967296	No
IEEFP	Single-precision floating point	No

Optimizing scanning performance

Siemens S5 scan packets contain up to 100 addresses.

Siemens S7 scan packets contain up to 19 addresses, due to limitations in packet size that can be sent to the S7 controller.

A Siemens S5 scan packet consists of a complete variable including all of its elements. All the elements of a variable are read in one request regardless of the number of elements required by the scan request. Therefore, a variable should be defined with elements all of the same scan period. In order to be in a scan packet, points must reside in one controller.

Since the fastest scan rate specified within a variable is always used for all points referencing the variable, it is important to match scan rates of the elements within a variable.

Try to make each packet as close to the maximum size as possible. Ensure there are no small packets being scanned at fast rates.

If more than one parameter references the same address in a controller, only one scan packet entry is created. If the parameters do not have the same scan rate, the scan entry will be created for the faster of the scan rates for those parameters.

If two points reference the same address but have different scan rates, the server only scans the address at the fastest scan rate.

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

Server tasks for Siemens

4

This chapter describes tasks for the Siemens controller that you perform either on the server or from any Station.

For information about:	Go to:
Testing communications with field devices	page 30
Troubleshooting point configuration errors	page 32

Testing Siemens communications

The diagnostic utility you use depends on the controller type.

Testing Siemens S5 controllers

You use the diagnostic utility, **smntst**, to test communications between the Siemens S5 controller and the server.

**Note**

Do not run **smntst** while the server is running because it will interfere with the operation of the server. (The system services must be stopped, but the server database and daemon services must be running.)

To run the utility:

- 1 Open a Command Prompt window.
- 2 Type **smntst** and press ENTER.
- 3 Follow the directions as prompted. Note that you should access menu items sequentially. (Warning messages are generated if a selected menu item is not currently available.)

Testing Siemens S7 controllers

You use the diagnostic utility, **sms7tstd**, to test communications between the Siemens S7 controller and the server, and to verify the Siemens variable addressing syntax—see “Supported symbolic addresses (S7)” on page 24. Note that the utility uses Siemens rather than server point names.

The utility does not make any assumptions regarding the format of points read from the controller. All values are shown in hexadecimal, decimal, and floating point representation. Depending on the point address, one or more of these may not be relevant. The length and error variables are values returned from the Siemens S7 API.

**Note**

Do not run **sms7tstd** while the server is running because it will interfere with the operation of the server. (The system services must be stopped, but the server database and daemon services must be running.)

To run the utility:

- 1 Open a Command Prompt window.
- 2 Type **sms7tstd** and press ENTER.
- 3 Follow the directions as prompted. Note that you should access menu items sequentially. (Warning messages are generated if a selected menu item is not currently available.)

Troubleshooting point configuration errors

Some things to check for both types of Siemens controller (S5 and S7) include:

- If the link is broken, error a042 will appear in log file. Plug the cable back in.
- The controller-CP interface is configured.
- A variable defined in the CP card is configured with a unique status word.
- The controller or CP card is online.
- The application association specified for an controller (LINK) has been defined in the CP database.

Troubleshooting S5 controllers

Two problems became evident during testing of the Siemens interface:

- Large variable arrays in **smntst**.

If an array of variables is too large to fit in a single Protocol Data Unit (PDU), an error is generated when trying to read or write elements of that array. For example, an array of 256 signed 16-bit integers combined with packet headers cannot be carried by a 500-byte PDU. The error number shown by **smntst** is 9110 011e, which corresponds to “Error with tf_read - User data too long”. This problem can be avoided by increasing the PDU size in the server and the controller.

- Changing the scan period.

If the scan period is changed while a Siemens channel is active, errors will be generated with the messages “0 entries in ddt record” and “error reading siemens rtu” in the server log file, and the controller and channel will fail. To recover from these errors, disable and re-enable the channel. To avoid these errors entirely, take the channel out of service while changing the scan period.

Troubleshooting point configuration errors on S7 controllers

Incorrectly configured points may reveal themselves in one of two ways.

- Errors while downloading Quick Builder point definitions to the server.

If points are configured with illegal configuration details this may cause problems when they are downloaded to the server. If this occurs read the `xxxx.out` file created and correct the errors.

- Errors when scanning

If points are built with variables that are not configured in the controller, they will not be reported as errors until the server attempts to acquire data from

those points. They will be evident by the point detail showing a bad value (indicated by inverse video). Error messages about these points are printed in the error log. If this occurs, the points should be checked for references to variables that are not configured within the controller.

If all the points built against a particular controller are bad and this controller fails, the configuration of this controller might be incorrect. Verify that the VFD and Connection are defined in the controller.

Diagnostic failures on S7 controllers

The server performs a diagnostic once a minute to confirm communications with the S7 controller. This diagnostic relies on the existence of the timer point “T1” in the controller. If this point is not configured, the diagnostic will fail and the associated channel will get errors and possibly fail. To prevent this, ensure that the S7 variable “T1” can be read from the controller using the Siemens Monitor-Modify Variables program included with the Step7 software.

Index

A

application associations 11
architecture 10

C

channel, defining 18
communications processor card 6
communications processor database 12
 S5 series 14
 S7 series 12
communications testing 30
configuring Siemens, steps for 5
controller definition for a Siemens controller 20

D

database, finalizing 15
defining a Siemens address for a point
 parameter 22
devices supported 6
documentation 6

M

Main properties for a Siemens channel 18
Main properties for a Siemens controller 20
memory hole 12

P

point parameter, defining an address 22

R

RAM, reserving memory 12
reserving memory 12

S

S5 controller
 application associations 11
 configuring 11
 configuring the communications processor
 database 14
 data formats 23
 location syntax 22
 supported devices 6
 testing 30
 troubleshooting 32
 variables 11
S7 controller
 configuring 11
 configuring the communications processor
 database 12
 data formats 26
 location syntax 24
 supported devices 6
 supported symbolic addresses 24
 symbolic addresses 14
 testing 30
 troubleshooting 32, 33
scanning, optimizing performance 27
server, setting up 12
setting up a Siemens controller 11
Siemens
 architecture 10
 setting up a controller 11
 setting up the server 12
 supported devices 6
 testing communications 30
 troubleshooting 32
supported devices 6
symbolic addresses 14

T

testing communications 30
troubleshooting 32

V

variables 11

Yamatake MA500 Interface Reference

Issue

4.1

Date

November 2005

Notice

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 Limited Australia.

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 2005 – Honeywell Limited Australia

Honeywell trademarks

PlantScape[®], SafeBrowse[®], **TotalPlant**[®] and TDC 3000[®] are U.S. registered trademarks of Honeywell International Inc.

Experion[™] and Honeywell Enterprise Buildings Integrator[™] are 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.

Support and other contacts

For technical assistance, call your nearest Honeywell office.

Contents

1	Getting started	5
	Support and documentation for MA500	6
	MA500 models supported	6
	Other documentation for MA500	6
	MA500-specific terms	7
2	MA500 controller setup	9
	Architectures	10
	Cabling	11
	Communication settings	12
3	MA500 controller configuration and addressing	13
	Defining a MA500 channel	14
	MA500 channel main properties	14
	MA500 port properties	15
	Defining a MA500 controller	18
	MA500 controller main properties	18
	Defining a MA500 address for a point parameter value	20
	Entering an address	20
	Optimizing scanning performance	22
	MA500 scan packets	22
4	Server and Station tasks for MA500	23
	Testing communications with the server	24
	Troubleshooting point configuration errors	25

Getting started

1

This reference describes how to set up, configure, and test MA500 controller communications with the server. There is detailed information for defining the controller using Quick Builder.

Complete each step before commencing the next.

Steps for connecting and configuring an MA500 controller.

Steps	Go to:
Set the communication parameters	page 12
Connect the cables from the server to the MA500 controller	
Define channels	page 14
Define controllers	page 18
Download channel and controller definitions to the server	
Test communications	page 24
Define points with Quick Builder	page 20

Support and documentation for MA500

MA500 models supported

The server supports only one device using this interface, the CNM7 (communication module).

Other documentation for MA500

The following book describes the address space of all the devices supported by this interface in great detail. It also provides information about the physical configuration of the Units. This book is recommended when using this interface.

- Yamatake-Honeywell Factory Automation Controller—*MA500 Host Communications User Manual*; Part No: CP-UM-1231E

The remaining manuals provide detailed information concerning the configuration of each type of Unit.

- Yamatake-Honeywell FA Controller—*MA500 Operation Manual Specifications and Installation*; Part No: CP-UM-1226E
- Yamatake-Honeywell FA Controller —*MA500 Operation Manual Programming*; Part No: CP-UM-1227E
- Yamatake-Honeywell FA Controller —*MA500 Operation Manual Personal Computer Loader*; Part No: CP-UM-1229E

MA500-specific terms

Master Station Number

The station address of the CNM7 module. This is a dial setting on the front of the CNM7 card.

MA500 controller setup

2

This chapter describes how to set up an MA500 controller for the system.
The tasks for setting up a MA500 controller are:

For	Go to:
Supported architectures	page 10
Connecting to the server	page 11
Communication settings	page 12

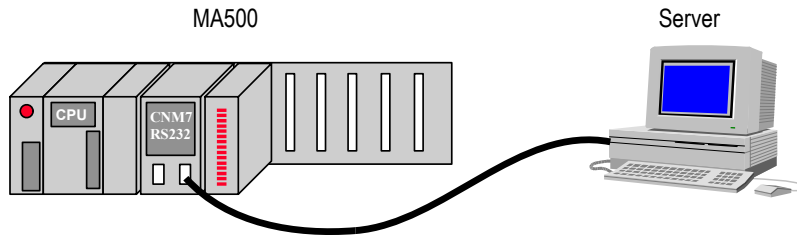
Architectures

MA500 PLCs are configurable in two different ways:

Standalone

A standalone system consists of only the base unit.

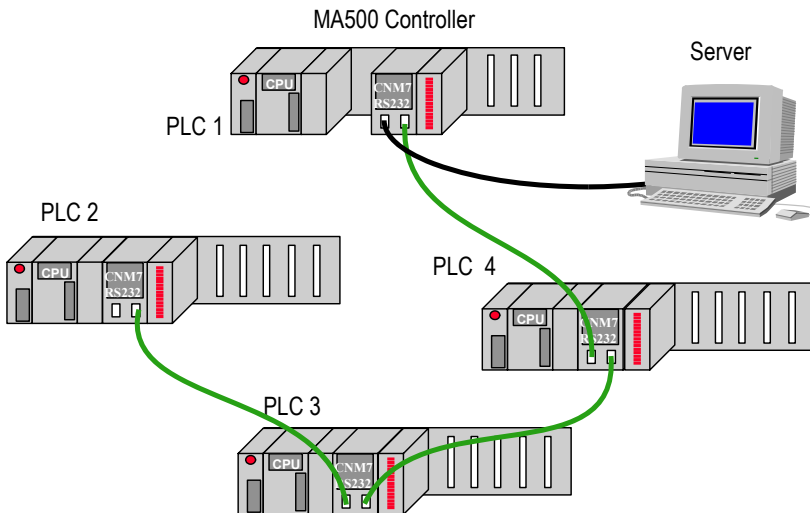
Figure 1 MA500 standalone system architecture



Networked

MA500 PLCs can be networked with the use of networking modules. Up to 63 MA500 PLCs can be networked in an ECL Network.

Figure 2 MA500 network architecture



The server communicates with the CNM7 in PLC1 via a serial RS-232 link.

The other MA500 PLCs are connected to PLC1's CNM7 via the ECL network.

The server can access points in all PLCs via the serial link by varying the Master Station Number in the MA500 read and write commands.

Multiple networking not supported

The server interface does not currently support a configuration having multiple CNM7 modules in an MA500, each CNM7 module having its own network with multiple PLCs on it.

Cabling

Connect the CNM7 to the server using RS-232 cables. The *MA500 Host Communications User Manual* provides the pin-out diagrams for RS-232 cabling.

Communication settings

The first step is to set up the MA500 controllers. See the manufacturers' documentation for details on setting up these parameters.

Communication parameters are detailed in the *MA500 Host Communications User Manual*. The following table details the specifications for the CNM7 module.

Item	Specification
Interface	RS-232 port
Baud Rate	2400, 4800, 9600, 19200
Communication	Half-Duplex
Transmission	Serial
Transmission code	ASCII
Error Control	parity, overrun, framing, checksum
Maximum message length	575 bytes/message (ECL limitation)

MA500 controller configuration and addressing

3

This chapter describes how to configure a MA500 controller for the system using Quick Builder. For each configuration procedure, there is detailed information covering all supported MA500 devices.

The Quick Builder controller configuration tasks are described:

For:	Go to:
Steps for defining a MA500 channel	page 14
Steps for defining a MA500 controller	page 18
Defining an address for a point parameter value	page 20
How to optimize scanning performance	page 22

Defining a MA500 channel

You need to define a channel for the Serial connection the server uses to communicate with the MA500 controllers.

To define a channel using Quick Builder:

- 1 Click  to add a channel.
- 2 In the Add Items dialog box, select channel as the item and MA500 as the channel type.
- 3 Enter the channel details on the Main property page for the channel. For help with the channel definitions, see MA500 channel main properties.
- 4 To complete the channel definition, click the Port tab and define a port. See “MA500 port properties” on page 15.

MA500 channel main properties

Use the Main tab to enter the basic channel properties for a MA500 channel.

Property	Description
Name	Type a unique name for the controller. A maximum of 10 alphanumeric characters can be used. Note: • Double quotation marks are not allowed. • For item names, spaces are not allowed.
Description (optional)	Type a description of the channel. A maximum of 30 characters can be used, including spaces.
Marginal Alarm Limit	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 2 or more, depending on the error and decrements for each good call. Type a new value in this field if you do not want the default, 25. To compute an acceptable limit, make it 25 times the square root of the number of controllers on the channel (for example, 9 controllers on channel [3 is square root] x 25 = 75).

Property	Description
Fail Alarm Limit	Communications alarm fail limit at which the channel is declared to have failed. When this limit is reached, an urgent 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 2 or more, depending on the error and decrements for each good call. Type a new value in this field if you do not want the default, 50. To compute an acceptable limit, double the Marginal Alarm Limit specified.
Connect Timeout	Amount of time, in seconds, the server will wait to connect to the controller before abandoning the connection. Type a new value in this field if you do not want the default, 10 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
Read Timeout	Amount of time, in seconds, that the server will wait for a reply from the controller. Type a new value in this field if you do not want the default, 2 seconds. Use the default value unless the communications line has a high error rate or unless you are using modems.
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.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field if you need to match your current server database configuration. The item number must be between 1 and the maximum number of channels allowed for your system.

MA500 port properties

The Port tab defines the communication-related properties for a channel. The **Port Type** for MA500 can be:

- **Serial.** A serial communications interface, such as RS-232. See “Serial port properties” on page 16.
- **TerminalServer.** Not applicable.
- **LANVendor.** Not applicable.

Serial port properties



Note

The Serial Port settings must match the settings on your communication devices.

Property	Description
Serial Port Name	The device name of the serial port.
Baud Rate	The number of data bits per second. The default is 9600.
Number of Data Bits	The number of data bits used for transmission. The default is 8.
Stop Bits	The number of stop bits used for transmission. The default is 1.
Parity	The parity verification used on the port. The default is NONE.
Checksum	The type of checksum error detection used for the port. Select the value that matches the setting on the communication device: • CRC16_0 or CRC16_1 (if Cyclic Redundancy Check (CRC) is set) • ONESCOMP or TWOSCOMP (if Longitudinal Redundancy Check (LRC) is set) • XOR (If exclusive or is set)
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: • None (default) • Input (use XON/XOFF to control the flow of data on the receive line) • output (use XON/XOFF to control the flow of data on the transmit line)

Property	Description
RS-232	Only applicable for a an RS-232 link. (The RS-232 and RS-485 settings are mutually exclusive.) 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.
RS-485	Only applicable for a an RS-485 link. (The RS-485 and RS-232 settings are mutually exclusive.) Enable Stallion RS-485 Half Duplex. Select this if you are using a Stallion EasyConnection. Do not select this if you use a 4-wire Stallion RS-422 connection. Echo. Select this to indicate that the server should expect the messages it sends to the port on the transmit line to be echoed back on the receive line. For example, select this if you are using a Stallion EasyConnection adapter. For RS-485 connections using a Black Box Converter, leave the box unselected (no echo).

Defining a MA500 controller

You need to define a controller to describe each PLC on a channel.

To define a controller using Quick Builder:

- 1 Click  to add a controller.
- 2 In the Add Items dialog box, select Controller as the item and MA500 as the controller type.
- 3 Enter the property definitions for the controller on the controller Main property page.

MA500 controller main properties

Use the Main tab to define the basic properties for a MA500 controller.

Property	Description
Name	Type a unique name for the controller. A maximum of 10 alphanumeric characters can be used. Note: • Double quotation marks are not allowed. • For item names, spaces are not allowed. • For LAN connected controllers, the name must not contain underscore (_) characters. This name is used to look up the IP address in the TCP/IP database if you do not specify an IP Address property.
Description (optional)	Type a meaningful description for this controller. A maximum of 30 characters, including spaces, can be entered.
Channel Name	The name of the channel on which the controller communicates. In the list of channel names, click the name. You need to have defined the channel in order for its name to appear in the list.

Property	Description
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this value is reached, a high priority alarm is generated. This limit applies to the controller barometer which monitors the total number of requests to the controller and the number of times the controller did not respond or response was incorrect. The barometer increments by 2 or more, depending on the error and decrements for each good call. Type a new value in this field if you do not want the default, 10. To determine an acceptable limit for this controller, make it 25 times the square root of the number of controllers on the channel (for example, 9 controllers on the channel $[3 \text{ is square root}] \times 25 = 75$).
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this value is reached, an urgent alarm is generated. Type a new value in this field if you do not want the default, 20. To determine an acceptable limit, double the marginal alarm limit value specified.
Station Number	Enter the station number of the MA500 controller, that is, the Master Station Number.
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.
Item Number	This field displays the unique item number currently assigned to this item by Quick Builder. You can change the item number displayed in this field 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.

Defining a MA500 address for a point parameter value

Entering an address

For **PV Source Address**, **Source Address**, and **Destination Address** the format for a MA500 controller address is:

ControllerName Location

Part	Description
<i>ControllerName</i>	The name of the MA500 controller.
<i>Location</i>	The location in the controller where the value is recorded. See “Location syntax” on page 20.

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

Location syntax

The syntax for the location is:

Address [Format | BitNumber]

Part	Description
<i>Address</i>	The byte to read from the PLC. The entire MA500 address space is accessible for point source addressing. There is a “special area” (250 to 265) which is read only and so unavailable for destination addresses. The Relay area and the Register area are available for destination addresses.
<i>Format</i>	(Optional) Valid for analog and accumulator points only. In the case of an analogue point, the data format must be defined. See “Data formats” on page 21.
<i>BitNumber</i>	(Optional) Valid for status points only. The valid range for bit number is 0 (default) to 15, where 0 is the right-most bit in the register.

Example Analog point example
 TIC-101 002 1000 U4095

 Status point example
 FCE-PRESS 002 202 6

 Accumulator point example
 PVSOURCE FCE-KWH 002 2001

Data formats

A data format is valid for analog and accumulator points only. Valid data format descriptors are shown below. The default format is C16.

Data Format	Description	Counts	Scaled
U4095	12-bit unsigned	0 to 4,095	yes
U16B	16-bit unsigned	0 to 65535	yes
S16B	16-bit signed	-32767 to 32768	yes
U15B	15-bit unsigned	0 to 32768	yes
C16	16-bit signed	-32767 to 32768	no

‘U’ range format types are scaled by the 0% and 100% of the HOST range values. ‘C’ types are not scaled and the 0% and 100% range values are used for adjusting the height of the display bar only.

Optimizing scanning performance

The maximum amount of data that can be acquired from a controller is influenced by the rate of sending scan packets to the controller. An understanding of the MA500 scan packets will help you configure points so that optimal data acquisition performance can be achieved by maximizing the amount of data acquired with each scan packet.

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. See the *Configuration Guide* for your server for usage of **lisscn**.

MA500 scan packets

An MA500 scan packet can consist of up to thirty two 16-bit words. It can be located anywhere along the full address range of the controller. In order to be in a scan packet, within the block, points must reside in the one controller and have the same scan period. In addition they must be addressed within the 32-word range that defines the block.

Scan packets can consist of up to 32 consecutive registers in any area of the address space. Registers can contain status, accumulator and analog points.

Types of scan packets built for an MA500 Interface:

Hardware diagnostic

One scan per controller every 60 seconds to verify communications integrity with the RTU.

Periodic data acquisition

One per scan packet enter the PV scan value for the desired rate at which the server should scan the MA500 PLC value. If it is a fast changing or critical parameter, use a fast rate. If the parameter changes infrequently or is non-critical, use a slower rate.

Server and Station tasks for MA500

4

This chapter describes tasks for the MA500 controller that you perform either on the server or from any Station.

For:	Go to:
Testing communications with the server	page 24
Troubleshooting point configuration errors	page 25

Testing communications with the server

An MA500 diagnostic utility, **ma5tst**, is included in the server software. After defining the channels and controller, you can test communication between the server and an MA500 PLC.

Before testing, complete the following:

- Set up the MA500 PLC.
- Connect all cables.
- Define the channels and controllers with Quick Builder and successfully download the definitions to the server.



Note

- If there is a Loader Terminal with the system, this should be used to ensure that MA500 is operating correctly.
 - The server need not be running while using the utility.
-

To run the utility:

- 1 Open a command prompt window and enter **ma5tst**.
- 2 When prompted, enter the number of a MA500 channel previously configured in Quick Builder. You will then be prompted with a menu of MA500 test functions. Select from this list of functions to test communication with the MA500 controller.
- 3 If any errors are encountered, review the channel and controller definitions.

Once it is verified that the server is communicating with the MA500 PLC, points can be built to reference MA500 addresses.

Troubleshooting point configuration errors

Incorrectly configured points may reveal themselves in one of two ways.

Errors while downloading to host.

If points are configured with illegal configuration details this may cause problems when they are downloaded to the host. If this occurs read the out file created and correct the errors.

Errors when scanning

If points are built with addresses which are valid but not configured in the Controller then these will not be reported as errors until the server attempts to acquire data from the points. They will be evident by the point detail showing a bad value (indicated by inverse video).

The most common errors are:

84E0 invalid address

Check that the address lies within the correct range. For MA500 this is 0 - 4095. Also check that an xxDESTIN has not been requested in the “special area” (250-265).

84E7 invalid bit number

Check that the bit number lies in the valid range (0 to 15). Also check that a read or write has not been requested across a word boundary.

For example the following statements will produce such an error:

Case 1

&

RANGE	MA5ERR1	0	7.0	
PVSOURCE	MA5ERR1	008	2000	14

&

Case 2

&

OPWIDTH	MA5ERR2	2		
OPDESTIN	MA5ERR2	008	2001	15

&

When this occurs the points should be checked for references to addresses in the Controller that are not configured within the Controller.

Index

A

addressing, MA500 PLC 20
architecture 10

B

baud rate, setting for server communication 12

C

cables, for CNM7 to server 11
channel definition
 for MA500 14
channels
 Port tab 15
configuration, for standalone and networked
 MA500 units 10
configuring MA500, steps for 5
controller
 addressing MA500 PLC 20
 definition in Quick Builder for MA500 18

D

data formats, for point 21
documentation 6

E

ECL maximum message length 12

M

Master Station number, definition 7

P

point
 data formats 21

errors in configuration 25
Port tab 15
Property Page
 Port tab 15

S

scan packets 22
scanning
 optimizing for MA500 22

Honeywell Process Solutions

2500 West Union Hills Drive

Phoenix AZ 85027

USA

www.honeywell.com

EP-DSX114

11/05

© 2005 Honeywell International Inc

Honeywell