Experion PKS

Honeywell Universal Modbus Interface Reference

EPDOC-XX64-en-431A
February 2015

Release 431

Honeywell

Document	Release	Issue	Date
EPDOC-XX64-en-431A	431	0	February 2015

Disclaimer

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

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

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

Contents

Planning considerations for installing and configuring Honeywell Universal Modbus controllers

This reference provides the information you need to set up, configure, and test Universal Modbus controller communications with the server.

Revision history

Revision	Date	Description
A	February 2015	Initial release of document.

How to use this guide

These are the steps for connecting and configuring a Honeywell Universal Modbus controller. Complete each step before commencing the next.

Steps	Go to
Connect and set up the Universal Modbus controller according to the controller's user manual's instructions	Architectures for Honeywell Universal Modbus
Use Quick Builder to define channels	• Honeywell Universal Modbus channel and controller reference • "Build channels" topic in the *Quick Builder User’s Guide*
Use Quick Builder to define controllers	• Honeywell Universal Modbus channel and controller reference • "Build controllers" topic in the *Quick Builder User’s Guide*

Steps	Go to
Download channel and controller definitions to the server	"Downloading items" topic in the *Quick Builder User's Guide*
Enable channels and test communications	Testing Honeywell Universal Modbus communications with the server
Troubleshoot communication errors	Troubleshooting Honeywell Universal Modbus configuration errors
Use Quick Builder to define points	Defining a Honeywell Universal Modbus address for a point parameter

Related topics

"About the Honeywell Universal Modbus interface" on page 9

"Other documentation for Honeywell Universal Modbus" on page 10

"Devices supported by the Honeywell Universal Modbus interface" on page 11

"Bauds supported by Honeywell Universal Modbus" on page 12

"Numbered addresses" on page 13

"Non-numbered addresses" on page 60

"Architectures for Honeywell Universal Modbus" on page 62

"Honeywell Universal Modbus channel and controller reference" on page 73

"Testing Honeywell Universal Modbus communications with the server" on page 122

"Troubleshooting Honeywell Universal Modbus configuration errors" on page 124

"Defining a Honeywell Universal Modbus address for a point parameter" on page 88

About the Honeywell Universal Modbus interface

The Universal Modbus Interface enables the server to interface to any Control Products controller that implements the Honeywell Universal Modbus protocol. The Honeywell Universal 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.

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

To check your system license

1. From the Station menu, select **Configure** > **Server License Details**.

   The **Server License Details** display appears.

2. Select the **Interfaces** tab.

   All licensed options for your system appear with a green LED. Ensure that **Universal Modbus** is listed.

   Contact your local supplier for further licensing details.

Other documentation for Honeywell Universal Modbus

The following documents are available from Honeywell:

- *Modbus RTU Serial Communications User Manual* (Part number 51-52-25-66)
- *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
DR4300	51-52-25-71
DR4500	51-52-25-69
UDC2300	51-52-25-75
UDC3300	51-52-25-70 51-52-25-38 UDC3000 A Modbus 485 RTU Communication Manual
DPR180/DPR250	EN1I-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, DPR180/DPR250)	51-52-25-96 Ethernet Interface Manual
X-Series Paperless Recorder	43-TV-25-30 Product Manual

Devices supported by the Honeywell Universal Modbus interface

This following devices are supported by the Universal Modbus controller:

- DR4300
- DR4500
- DPR180
- DPR250
- UDC2300
- UDC3300
- UMC800
- HC900
- Trendview/X-Series

Bauds supported by Honeywell Universal Modbus

The following table lists the devices and their supported bauds.

! **Attention**

Bauds are not applicable to HC900 and TrendView devices; these devices use Ethernet connections.

Device	Bauds Supported							
	300	600	1200	2400	4800	9600	19,200	38,400
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

The section describes numbered addresses for Honeywell Universal Modbus controllers.

Related topics

"PID loop" on page 14
"Analog input" on page 19
"Tagged signal" on page 20
"Communication or constant value group" on page 21
"Math variable or calculated value group" on page 21
"Totalizer value group" on page 22
"Alarm set point value group" on page 23
"Math variable or calculated value status" on page 24
"Totalizer value status" on page 24
"Alarm status" on page 25
"Alarm status analog" on page 26
"Alarm status digital" on page 26
"Alarm status event" on page 27
"Alarm status channel" on page 27
"Alarm status communications" on page 28
"Alarm status math" on page 29
"Digital input table" on page 29
"Digital output table" on page 30
"Set point programmer" on page 31
"Set point program additional values" on page 36
"Set point programmer #1 profile segment" on page 37
"Set point programmer #2 profile segment" on page 39
"Set point programmer #3 profile segment" on page 42
"Set point programmer #4 profile segment" on page 44
"Set point scheduler values" on page 47
"Set point scheduler #1 segment" on page 51
"Set point scheduler #2 segment" on page 55
"Address syntax for named addresses" on page 89

PID loop

Devices supported

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

Device	Supported Address Format	Range
HC900	LOOP [n] [param]	[n] = 1 to 24
HC900	LOOPX [n] [param]	[n] = 25 to 32
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

Parameters

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

Param	Address Line	Param Format	Access	Devices
Active/Inactive LO	LOOP [n] STATUS_LO	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Alarm #1 SP #1	LOOP [n] AL1SP1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Alarm #1 SP #2	LOOP [n] AL1SP2	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Alarm #2 SP #1	LOOP [n] AL2SP1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Alarm #2 SP #2	LOOP [n] AL2SP2	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, 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

Param	Address Line	Param Format	Access	Devices
Auto/Manual	LOOP [n] AMSTAT	Discrete (bits). [Status Point Only]	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
BIAS	LOOP [n] BIAS	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Carbon Potential Dewpoint	LOOP [n] CPD	Floating Point	RW	UMC800, HC900
Currently Selected Local or Remote Set Point	LOOP [n] STATUS_RSP	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Currently Selected Set Point	LOOP [n] STATUS_SP	Discrete (bits). [Status Point Only]	RO	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	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Cycle Time #1	LOOP [n] CYCLE1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300
			RO	UMC800, HC900
Cycle Time #2	LOOP [n] CYCLE2	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300
			RO	UMC800, HC900
DB	LOOP [n] DB	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Demand Tune Request	LOOP [n] TUNE_REQ	Discrete (bits). [Status Point Only] Bit 0, 0 = Off, 1 = On	RW	UMC800, HC900
Deviation	LOOP [n] DEV	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
DIR	LOOP [n] DIR	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Feed Forward Gain	LOOP [n] FF_GAIN	Floating Point	RW	UMC800, HC900
Furnace Factor	LOOP [n] FFCTR	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
Gain #1 (Prop Band #1 if active)	LOOP [n] GAIN1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Gain #2 (Prop Band #2 if active)	LOOP [n] GAIN2	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
IMAN Active/ Inactive	LOOP [n] STATUS_IMAN	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Input #1	LOOP [n] INP1	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Input #2	LOOP [n] INP2	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300
Local Percent Carbon Monoxide	LOOP [n] PCTCO	Floating Point	RW	UMC800, HC900
Local Set Point #1	LOOP [n] LSP1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Local Set Point #2	LOOP [n] LSP2	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Local Set Point #3	LOOP [n] LSP3	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300
Manual Reset	LOOP [n] MAN_RESET	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
OP High Limit	LOOP [n] OPHIGH	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Output	LOOP [n] OP	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Output Low Limit	LOOP [n] OPLOW	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Output Override Value	LOOP [n] OPOVR	Floating Point	RW	UDC2300, UDC3300
Output Working Value	LOOP [n] OPWORK	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Percent Hydrogen	LOOP [n] H2	Floating Point	RW	UMC800, HC900
PV High Range	LOOP [n] PVHIGH	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
PV Low Range	LOOP [n] PVLOW	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Process Variable	LOOP [n] PV[1]	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Process Variable Override Value	LOOP [n] PVOVR	Floating Point	RW	UDC2300, UDC3300
Prop Band #1	LOOP [n] PROP1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Prop Band #2	LOOP [n] PROP2	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate #1	LOOP [n] RATE1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
RATIO	LOOP [n] RATIO	Floating Point	RW	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	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

[1] The default parameter if only 'LOOP [n]' is entered.

Param	Address Line	Param Format	Access	Devices
Read-only mode for the PID Loop	LOOP [n] LOOPSTAT	Mode Status - Bit 0 = Auto/Man State. Bit 2 = LSP/RSP State.	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Remote/Local Set Point State	LOOP [n] RSP_STATE	Discrete (bits). [Status Point Only] Bit 0, 0 = LSP, 1 = RSP	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Remote Set Point (RSP)	LOOP [n] RSP	Floating Point	RW	UMC800, HC900
			RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Reset #1	LOOP [n] RESET1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Reset #2	LOOP [n] RESET2	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate #2	LOOP [n] RATE2	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point #1	LOOP [n] SP1	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point #2	LOOP [n] SP2	Floating Point	RW	UMC800, HC900
			RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Override Value	LOOP [n] SPOVR	Floating Point	RW	UDC2300, UDC3300
Set Point State	LOOP [n] SP_STATE	Discrete (bits). [Status Point Only]	RW	UDC2300, UDC3300
		Discrete (bits). [Status Point Only] Bit 0, 0 = SP1, 1 = SP2	RW	DR4300, DR4500, UDC2300, UMC800, HC900
SP Low Limit	LOOP [n] SPLOW	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
SP High Limit	LOOP [n] SPHIGH	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Temperature in carbon potential loop	LOOP [n] TEMP	Floating Point	RO	UMC800, HC900
Three Position Step Motor Time	LOOP [n] MOTOR	Floating Point	RW	UMC800, HC900
Tune Set State	LOOP [n] TUNE_SET_STATE	Discrete (bits). [Status Point Only]	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Working Set Point (SPWORK)	LOOP [n] WSP	Floating Point	RO	DR4300, DR4500
			RW	UDC2300, UDC3300, UMC800, HC900
Working Set Point (WSP)	LOOP [n] SPWORK	Floating Point	RW	UMC800, HC900
			RO	DR4300, DR4500, UDC2300, UDC3300
Writable Controller Mode	LOOP [n] MODEIN	Control Mode - Auto/Man State (bit 0) and LSP/RSP State (bit 2).	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Analog input

Devices supported

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

Device	Supported Address Format	Range
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

Device	Supported Address Format	Range
HC900	AI [n] [param]	[n] = 1 to 64[2]
TrendView	AI [n] [param]	[n] = 1 to 32

Parameters

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

Param	Address Line	Param Format	Access	Devices
Analog Input Value	AI [n] VALUE[3]	Floating Point	RO	DR4300, DR4500, DPR180, DPR250, UDC2300, UDC3300, UMC800, HC900, TV

Tagged signal

Devices supported

The following table lists the devices that 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 1,000

Parameters

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

Param	Address Line	Param Format	Access	Devices
Tagged Signal Value	TAG [n] VALUE[4]	Floating Point	RO	UMC800, HC900

[2] In 1st rack, first 8 slots.

[3] The default Parameter if only 'AI [n]' is entered.

[4] The default Parameter if only 'TAG [n]' is entered.

Communication or constant value group

Devices supported

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

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

Parameters

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[5]	Floating Point	RW	DPR180, DPR250, TV[6]

Math variable or calculated value group

Devices supported

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

Device	Supported Address Format	Range
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

[5] The default parameter if only 'CN [n]' is entered.

[6] Trendview/X-Series communication values are written via this parameter, read via Math/ Calculated values parameter (the 'pen' value).

Device	Supported Address Format	Range
UMC800	MATH_VAR [n] [param]	[n] = 1 to 150
HC900	MATH_VAR [n] [param]	[n] = 1 to 2048
TrendView	MATH_VAR [n] [param]	[n] = 1 to 64

Parameters

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

Param	Address Line	Param Format	Access	Devices
Math or Calculated Value	MATH_VAR [n] VALUE[7]	Floating Point	RO	DR4500, DPR180, DPR250, UDC3300, TV[8]
Math or Calculated Value	MATH_VAR [n] VALUE [7]	Floating Point	RW	UMC800[9], HC900 [9]

Totalizer value group

Devices supported

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

Device	Supported Address Format	Range
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

Parameters

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

[7] The default Parameter if only 'MATH_VAR [n]' is entered.

[8] Applies to TrendView or X-Series 'pen' values.

[9] Applies to UMC800, HC900 'variable' values.

Param	Address Line	Param Format	Access	Devices
Totalizer Value	TOTALIZER [n] VALUE	Floating Point	RW	DR4300, DR4500, UDC3300

Alarm set point value group

Devices supported

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

Device	Supported Address Format	Range
DPR180	ALMSP [n] [param]	[n] = 1 to 48
DPR250	ALMSP [n] [param]	[n] = 1 to 64
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

Parameters

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[10]	Floating Point	RW	DPR180, DPR250
Alarm Set Point #1	ALMSP [n] SP1[11]	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300
Alarm Set Point #2	ALMSP [n] SP2 [11]	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300

[10] The default Parameter if only 'ALMSP [n]' is entered.

[11] The default Parameter if only 'ALMSP [n]' is entered.

Math variable or calculated value status

Devices supported

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

Device	Supported Address Format	Range
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

Parameters

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[12]	Discrete (bits). [Status Point Only]	RO	DR4500, DPR180, DPR250, UDC3300

Totalizer value status

Devices supported

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

Device	Supported Address Format	Range
DR4300	TOTALIZER_STATUS [n] [param]	[n] = 1 to 1
DR4500	TOTALIZER_STATUS [n] [param]	[n] = 1 to 4
UDC3300	TOTALIZER_STATUS [n] [param]	[n] = 1 to 1

[12] The default Parameter if only 'MATH_STATUS [n]' is entered.

Parameters

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[13]	Discrete (bits). [Status Point Only] 0 = Totalizer Off 1 = Totalizer On	RO	DR4300, DR4500, UDC3300

Alarm status

Devices supported

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

Device	Supported Address Format	Range
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

Parameters

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

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT [n] STATUS[14]	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

[13] The default Parameter if only 'TOTALIZER_STATUS [n]' is entered.

[14] The default Parameter if only 'ALMSTAT [n]' is entered.

Alarm status analog

Devices supported

The following table lists the devices that 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

Parameters

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[15]	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

Alarm status digital

Devices supported

The following table lists the devices that 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

Parameters

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

[15] The default Parameter if only 'ALMSTAT_ANALOG [n]' is entered.

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT_DIGITAL [n] STATUS[16]	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

Alarm status event

Devices supported

The following table lists the devices that support the Alarm Status Event, 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

Parameters

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[17]	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

Alarm status channel

Devices supported

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

[16] The default Parameter if only 'ALMSTAT_DIGITAL [n]' is entered.

[17] The default Parameter if only 'ALMSTAT_EVENT [n]' is entered.

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

Parameters

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[18]	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

Alarm status communications

Devices supported

The following table lists the devices that support the Alarm Status Communications, 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

Parameters

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

Param	Address Line	Param Format	Access	Devices
Alarm Status	ALMSTAT_COM [n] STATUS[19]	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

[18] The default Parameter if only 'ALMSTAT_CHANNEL [n]' is entered.

[19] The default Parameter if only 'ALMSTAT_COM [n]' is entered.

Alarm status math

Devices supported

The following table lists the devices that 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

Parameters

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[20]	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250

Digital input table

Devices supported

Attention

Digital Input values reside in table 1. Therefore, they are not applicable for controllers with a **Data Table** setting of `Holding Register`.

The following table lists the devices that 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

[20] The default Parameter if only 'ALMSTAT_MATH [n]' is entered.

Device	Supported Address Format	Range
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

Parameters

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[21]	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250
Digital Input Value	DI [n] VALUE[22]	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC3300, UMC800, HC900[23], TV

Digital output table

Devices supported

Attention

Digital Output values reside in table 0. Therefore, they are not applicable for controllers with a **Data Table** setting of `Holding Register`.

The following table lists the devices that 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

[21] The default Parameter if only 'DI [n]' is entered.

[22] The default Parameter if only 'DI [n]' is entered.

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

Device	Supported Address Format	Range
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

Parameters

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[24]	Discrete (bits). [Status Point Only]	RO	DPR180, DPR250
Digital Output Value	DO [n] VALUE[25]	Discrete (bits). [Status Point Only]	RW	DR4300, DR4500
Digital Output Value	DO [n] VALUE [25]	Discrete (bits). [Status Point Only]	RO	UDC2300, UDC3300, UMC800, HC900[26], TV

Set point programmer

Devices supported

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

Device	Supported Address Format	Range
DR4300	SPP [n] [param]	[n] = 1 to 1

[24] The default Parameter if only 'DO [n]' is entered.

[25] The default Parameter if only 'DO [n]' is entered.

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

Device	Supported Address Format	Range
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

Parameters

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[27]	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Segment Time Remaining	SPP [n] SEG_TIME_REM	Floating Point	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Start	SPP [n] START	UINT2	WO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Hold	SPP [n] HOLD	UINT2	WO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Advance	SPP [n] ADV	UINT2	WO	UMC800, HC900
Set Point Programmer Reset	SPP [n] RESET	UINT2	WO	UMC800, HC900
Set Point Programmer Status - Ready	SPP [n] STATUS_READY	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - Run	SPP [n] STATUS_RUN	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

[27] The default Parameter if only 'SPP [n]' is entered.

Param	Address Line	Param Format	Access	Devices
Set Point Programmer Status - Hold	SPP [n] STATUS_HOLD	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - End	SPP [n] STATUS_END	Discrete (bits). [Status Point Only]	RO	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Status - Time Units in Seconds	SPP [n] STATUS_TIME_UNIT_S	Discrete (bits). [Status Point Only]	RO	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	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	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Set Point Programmer Elapsed Time	SPP [n] EL_TIME	Floating Point	RO	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	DR4300, DR4500, UDC2300, UDC3300
Set Point Programmer Current Segment Number	SPP [n] SEG_NO	Floating Point	RW	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Set Point Programmer Status - Type of Hold	SPP [n] STATUS_HOLD_TYPE	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
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	UMC800, HC900
Set Point Programmer Segment Event #1	SPP [n] EV01	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #2	SPP [n] EV02	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #3	SPP [n] EV03	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #4	SPP [n] EV04	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #5	SPP [n] EV05	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #6	SPP [n] EV06	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #7	SPP [n] EV07	Discrete (bits). [Status Point Only]	RO	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Set Point Programmer Segment Event #8	SPP [n] EV08	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #9	SPP [n] EV09	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #10	SPP [n] EV10	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #11	SPP [n] EV11	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #12	SPP [n] EV12	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #13	SPP [n] EV13	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #14	SPP [n] EV14	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #15	SPP [n] EV15	Discrete (bits). [Status Point Only]	RO	UMC800, HC900
Set Point Programmer Segment Event #16	SPP [n] EV16	Discrete (bits). [Status Point Only]	RO	UMC800, HC900

Set point program additional values

Devices supported

The following table lists the devices that 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

Parameters

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[28]	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

[28] The default Parameter if only 'SPP_ADD [n]' is entered.

Param	Address Line	Param Format	Access	Devices
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

Set point programmer #1 profile segment

Devices supported

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

Device	Supported Address Format	Range
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

Parameters

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
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

Param	Address Line	Param Format	Access	Devices
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
Soak Value for Auxiliary Output	SPP1_SEG [n] AUX_OUT	Floating Point	RW	UMC800, HC900
Time	SPP1_SEG [n] TIME[29]	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate	SPP1_SEG [n] RATE	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Ramp or Soak Value	SPP1_SEG [n] SEG_VALUE	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Set point programmer #2 profile segment

Devices supported

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

[29] The default Parameter if only 'SPP1_SEG [n]' is entered.

Device	Supported Address Format	Range
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

Parameters

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

Param	Address Line	Param Format	Access	Devices
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
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[30]	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Param	Address Line	Param Format	Access	Devices
Rate	SPP2_SEG [n] RATE	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Ramp or Soak Value	SPP2_SEG [n] SEG_VALUE	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Set point programmer #3 profile segment

Devices supported

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

Device	Supported Address Format	Range
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

Parameters

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

[30] The default Parameter if only 'SPP2_SEG [n]' is entered.

Param	Address Line	Param Format	Access	Devices
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
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

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

Set point programmer #4 profile segment

Devices supported

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

Device	Supported Address Format	Range
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

[31] The default Parameter if only 'SPP3_SEG [n]' is entered.

Device	Supported Address Format	Range
UMC800, HC900	SPP4_SEG [n] [param]	[n] = 1 to 50

Parameters

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
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

Param	Address Line	Param Format	Access	Devices
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
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[32]	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900
Rate	SPP4_SEG [n] RATE	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

[32] The default Parameter if only 'SPP4_SEG [n]' is entered.

Param	Address Line	Param Format	Access	Devices
Ramp or Soak Value	SPP4_SEG [n] SEG_VALUE	Floating Point	RW	DR4300, DR4500, UDC2300, UDC3300, UMC800, HC900

Set point scheduler values

Devices supported

The following table lists the devices that 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

Parameters

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

Param	Address Line	Param Format	Access	Devices
Output #1	SCHED [n] OUTPUT1[33]	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

[33] The default Parameter if only 'SCHED [n]' is entered.

Param	Address Line	Param Format	Access	Devices
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
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

Param	Address Line	Param Format	Access	Devices
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
Event #6	SCHED [n] EVENT_06	Discrete (bits). [Status Point Only]	RO	UMC800, HC900

Param	Address Line	Param Format	Access	Devices
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

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

Set point scheduler #1 segment

Devices supported

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

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

Parameters

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[34]	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

[34] The default Parameter if only 'SCHED1_SEG [n]' is entered.

Param	Address Line	Param Format	Access	Devices
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
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

Param	Address Line	Param Format	Access	Devices
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
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

Param	Address Line	Param Format	Access	Devices
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
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

Param	Address Line	Param Format	Access	Devices
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

Set point scheduler #2 segment

Devices supported

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

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

Parameters

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[35]	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

[35] The default Parameter if only 'SCHED2_SEG [n]' is entered.

Param	Address Line	Param Format	Access	Devices
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
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

Param	Address Line	Param Format	Access	Devices
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

Param	Address Line	Param Format	Access	Devices
Recycle Segment	SCHED2_SEG [n] RECYCLE_SEG	Floating Point	RW	HC900

Non-numbered addresses

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

Parameter	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

Parameter	Address Line	Param Format	Access	Devices
Alarm Relay #1	ALMRLY1	Discrete (bits). [Status Point Only]	RO	UDC2300, UDC3300
INSTMODE	INSTMODE	Floating Point	RW	UMC800, HC900
CONFIG_CLEAR	CONFIG_CLEAR	Floating Point	WO	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

Architectures for Honeywell Universal Modbus

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

Honeywell Universal Modbus connections

Control Products controllers are designed to communicate using the Modbus TCP Ethernet, RS485, or 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 the section below titled "Using an RS-232/RS-485 converter")
- Directly connecting the server to the RS-485 network via an add-in card (see the section below titled "Using an RS-485 adapter")

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

- Direct Ethernet Connection (HC900, single or redundant networks, Trendview/X-Series Paperless Recorder)
- Ethernet - Modbus bridge (internal option for UMC800, 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/RS-485 converter

Honeywell recommends that you use the Black Box LD485-HS RS-232/RS-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.

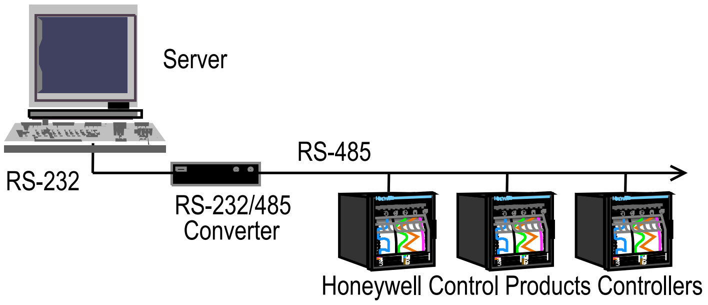

Figure 1: RS-232 to RS-485 converter

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 “Figure 2: Black Box (2-wire) connections”.

Black Box connections

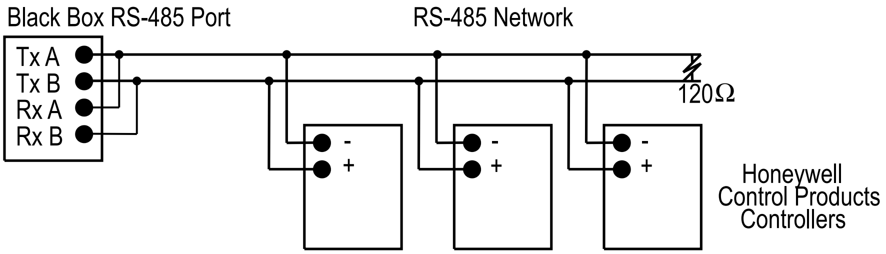

Figure 2: Black Box (2-wire) connections

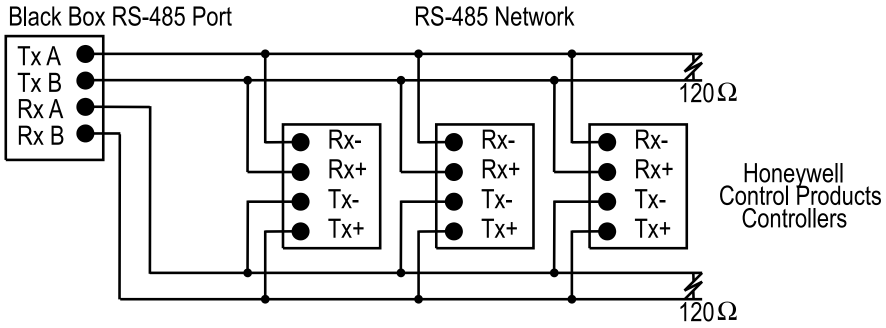

Figure 3: Black Box (4-wire) Connections

Ensure that the Black Box switches are configured with the following settings.

An asterisk (*) designates the factory-preset jumper 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; increase for lower baud. 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.

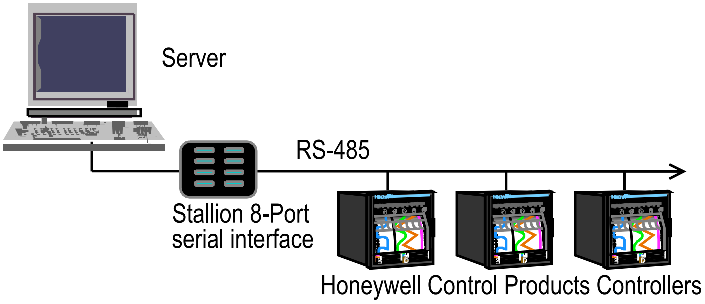
Server
RS-485
Stallion 8-Port serial interface
Honeywell Control Products Controllers

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 “Figure 5: Stallion RS-485 (2-wire) connections” and in “Figure 6: Stallion RS-485 (4-wire) connections”. Next, connect your Control Products controllers to the RS-485 network.

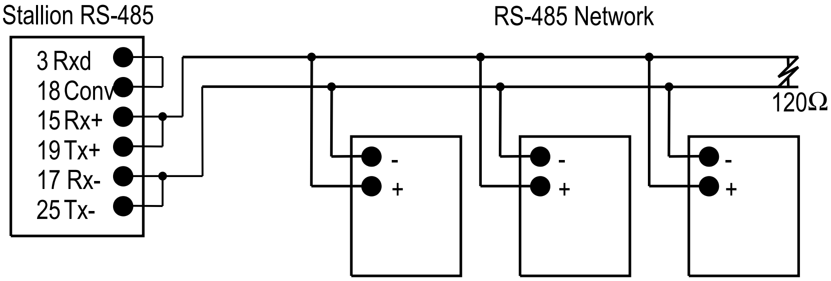

Figure 5: Stallion RS-485 (2-wire) connections

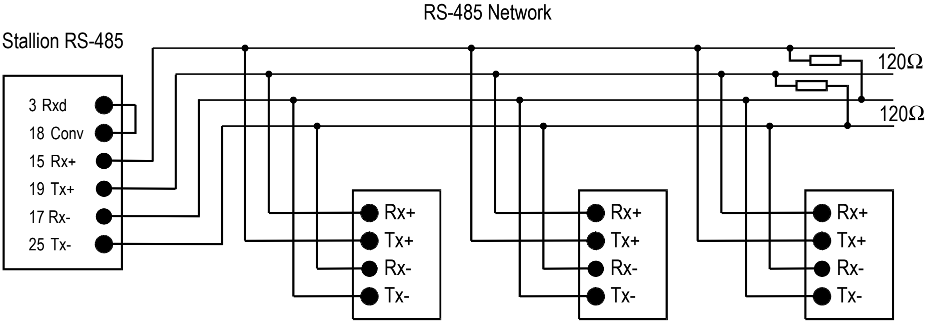

Figure 6: Stallion RS-485 (4-wire) connections

TCP connection for Honeywell Universal Modbus

To connect controllers to the server communicating using the Universal Modbus TCP protocol, you are required to have a *network interface card* (NIC) connected to an Ethernet network on both the server as well as the controller. An external TCP/IP bridge (Lantronix DR1-IAP) can also be used for RS-485 network connection to Ethernet.

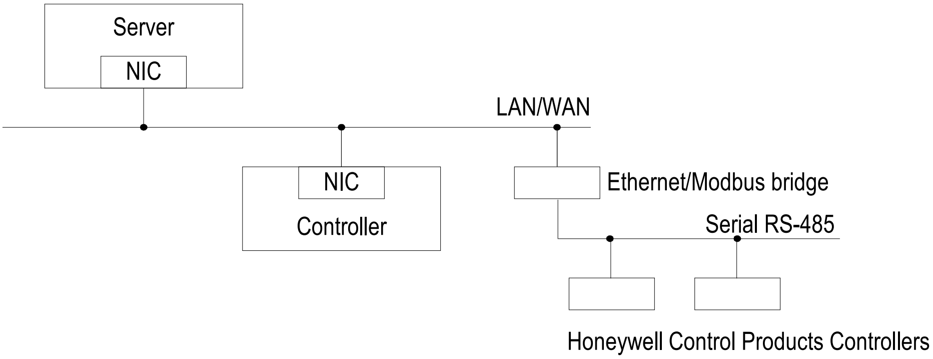

Figure 7: Non-redundant Universal Modbus TCP Network Architecture

C70 and C70R redundant communication and redundant controller architectures

The C70 and C70R models of the HC900 allow for redundant communication, and the C70R also allows for redundant controllers.

The following sections outline common architectures that involve these features. While a Backup Server is shown in each of these diagrams, the Backup Server is not required, and is included to show how a Backup Server would be connected if one existed.

Redundant controller

You are required to have a *network interface card* (NIC) on each server, and the E1 port on both controllers, connected to an Ethernet network via a switch.

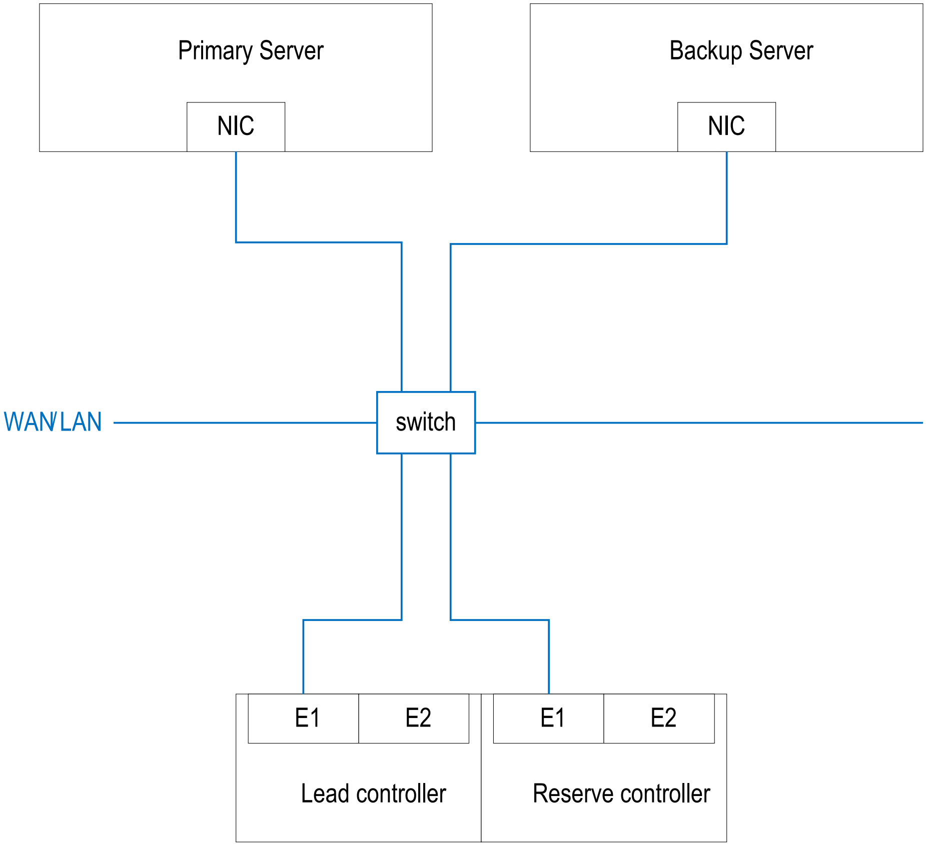

Figure 8: Redundant controller Universal Modbus TCP network architecture

Redundant controller and network

This is the same as the *Redundant controller* architecture, except that you require a second network adapter on each server, and the E2 port on both controllers, to be connected to a separate network (designated as LAN/WAN #2 in “Figure 9: Redundant controller and network Universal Modbus TCP network architecture”) via a second switch.

LAN/WAN #2 must be configured as a different subnet.

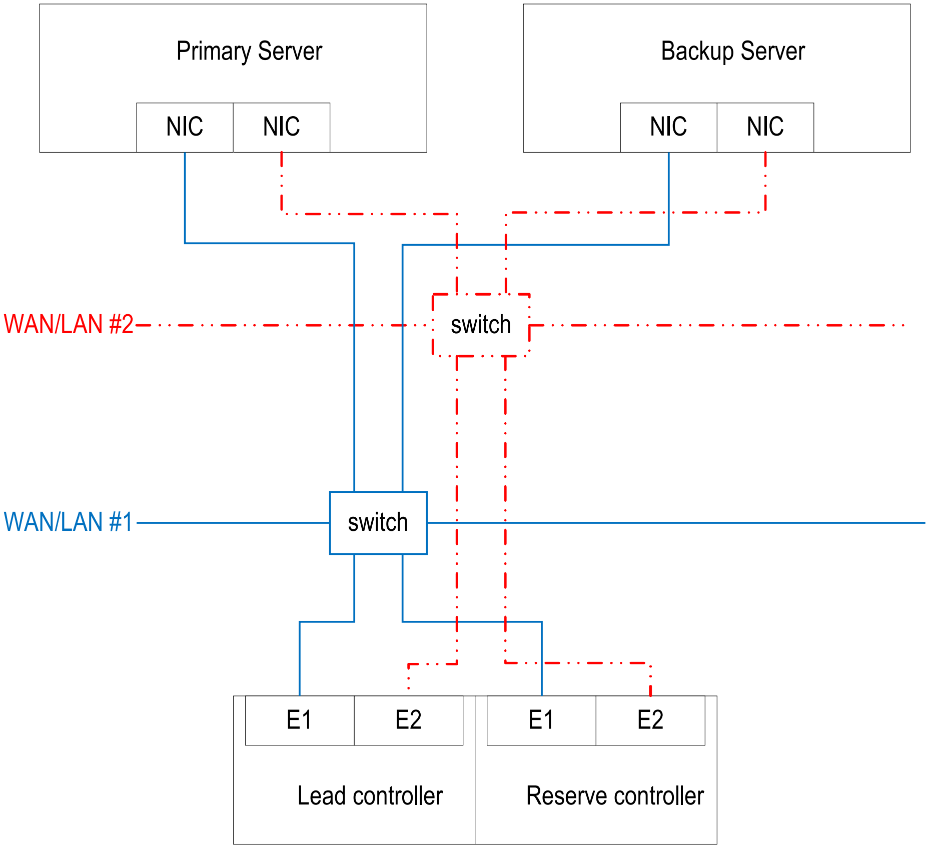

Figure 9: Redundant controller and network Universal Modbus TCP network architecture

Redundant controller on FTE

When connecting a HC900 to an FTE network with redundant controllers, there must be two network connections on each server connected via FTE switches to the FTE network. The E1 port on the HC900s must be connected to the FTE Yellow switch.

The ports on the FTE switch must be configured to:

- Block multicast and broadcast traffic
- Allow Auto-Detect or Half-Duplex (HC900 does not support Full Duplex communications)

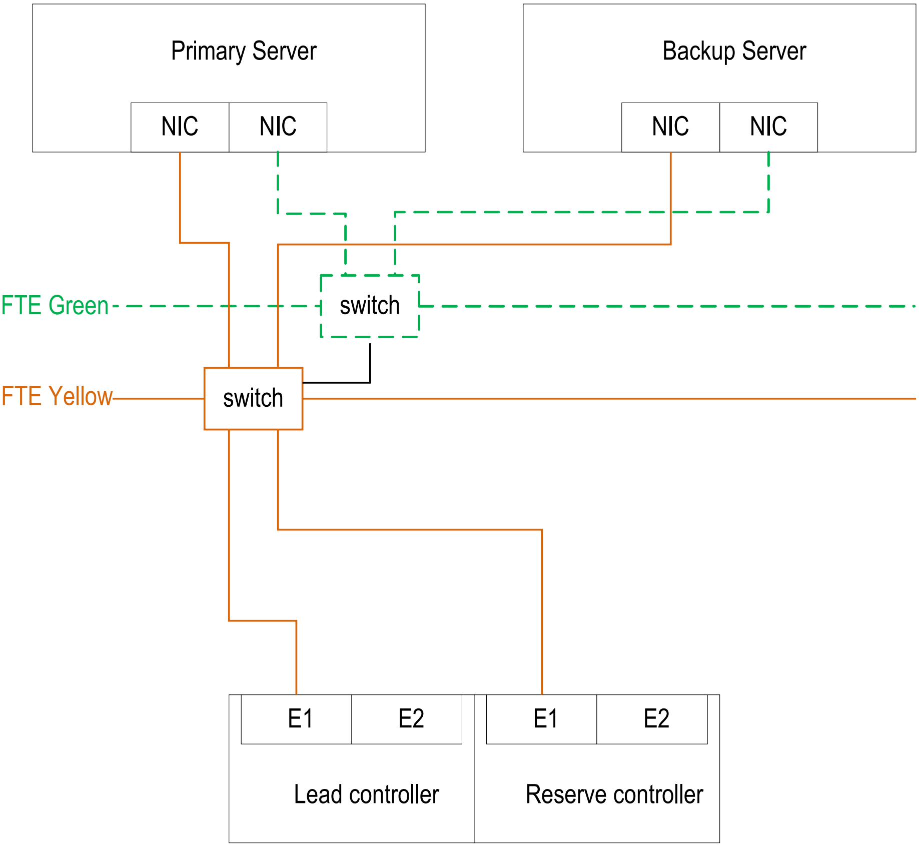

Figure 10: Redundant controller on FTE Universal Modbus TCP network architecture

Redundant controller and network on FTE

This is the same as the *Redundant controller on FTE* architecture, except that you require a third network adapter on each server, and the E2 port on both controllers, to be connected to a separate network (designated as Non-FTE LAN in “Figure 11: Redundant controller and network on FTE Universal Modbus TCP network architecture”) via a third switch.

Non-FTE LAN must be configured as a different subnet.

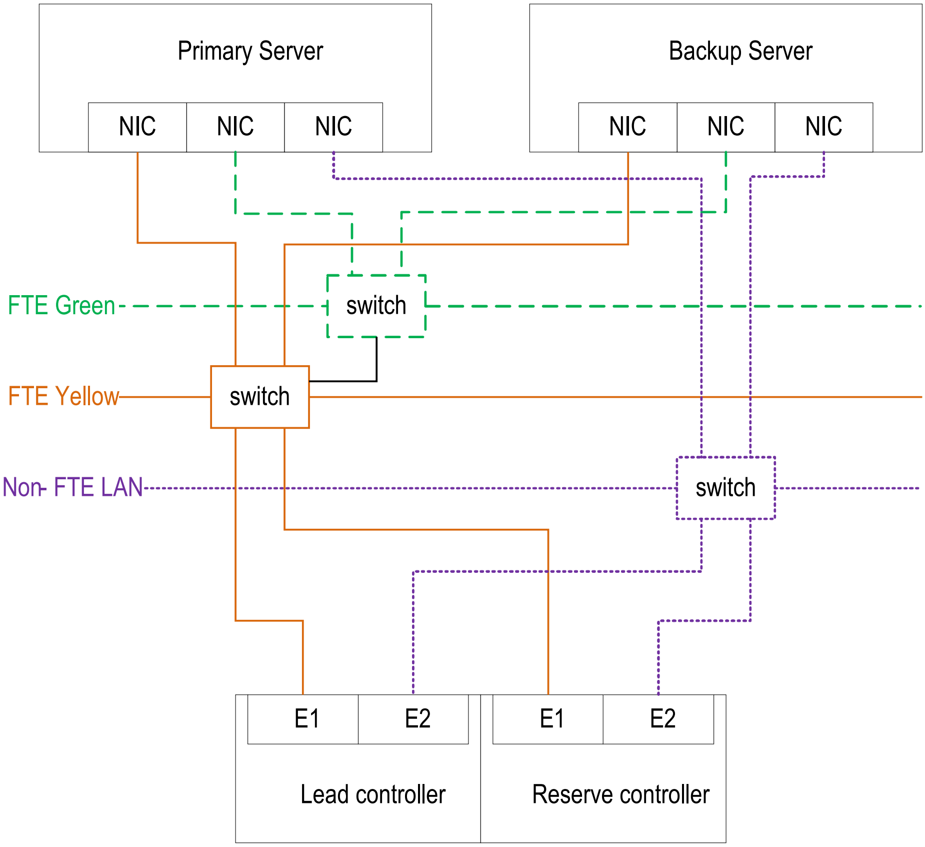

Figure 11: Redundant controller and network on FTE Universal Modbus TCP network architecture

RS-485 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. When you are ready to configure the server, you will need to know at what baud each RS-485 network is using.

Every controller using the same connection to the server (RS-232 or RS-485) should have a unique Universal 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 RS-485 devices. Each must also have a unique physical address on the RS-485 network.

Honeywell Universal Modbus channel and controller reference

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

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

Related topics

“Main properties for a Honeywell Universal Modbus channel” on page 74

“Port properties for a Honeywell Universal Modbus channel” on page 77

“Redundant port properties for a Honeywell Universal Modbus channel” on page 79

“Main properties for a Honeywell Universal Modbus controller” on page 80

“Setting up time synchronization for UMC800 and HC900 controllers” on page 85

“Planning considerations for installing and configuring Honeywell Universal Modbus controllers” on page 7

Main properties for a Honeywell Universal Modbus channel

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

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

Property	Description
Name	The unique name of the channel. A maximum of *10* alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces.
Description	(Optional) A description of the channel. A maximum of *132* alphanumeric characters, including spaces.
Associated Asset	The Tag Name of the Asset to be associated with the alarm group.
Marginal Alarm Limit	The communications alarm marginal limit at which the channel is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A channel barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. To calculate an acceptable marginal alarm limit, use the formula: Square root of the number of controllers on the channel × Marginal Alarm Limit defined on those controllers (Normally, you specify the same value for all controllers on a channel). For example, if there are 9 controllers on the channel and their Marginal Alarm Limit is set to *25*, the value would be *3* (which is the square root of 9) *x 25 = 75*.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the channel is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one channel, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the channel Marginal Alarm Limit.
Write Delay	If the channel is on a serial port, the length of time (in milliseconds) that the server waits before writing to any controller on the channel. See the section below titled "Channel write delay settings." For redundant HC900 networks, the suggested Write Delay is `0`.
Connect Timeout	The length of time that the server attempts to connect to the controller. The server will stop trying to connect to the controller once the timeout period passes. The default value `10` seconds. Use the default value unless the communications line has a high error rate, or unless you are using modems. For redundant HC900 networks, the suggested Connect Timeout is `1`.
Read Timeout	The length of time that the server will wait for a reply from the controller. The server will stop waiting once the timeout period passes. The default value is `2` seconds. Use the default value unless the communications line has a high error rate, or unless you are using modems. For redundant HC900 networks, the suggested Read Timeout is `1`.
Item Type	The type of channel specified when this item was created.
Last Modified	The date and time the channel properties were modified.
Last Downloaded	The date and time the channel was last downloaded to the server.
Item Number	The unique item number currently assigned to this channel, in the format `CHNcc`, where `cc` is the channel number. You can change the item number if you need to match your current server database configuration. The number must be between `01` and the maximum number of channels allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

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:

```
Time(ms) = (1,000 × characters)/baud
```

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

For example, 3.5 chars at 9600 baud = (1,000 × 3.5)/9600 = 3.6 ms (round to 4 ms)

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

Port properties for a Honeywell Universal Modbus channel

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

- `Serial`. A serial communications interface, such as RS-485.
- `TerminalServer`. A communications link that enables controllers with a serial interface to be connected to a LAN.
- `LANVendor`. A communications interface using TCP/IP to controllers connected to a LAN.

Serial port properties

Attention
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	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	Defines parity verification of each character and must match configuration on the end device. The default is `NONE`.
Checksum	The type of checksum error detection used for the port. Not applicable for this channel. Select `NONE`.
XON/XOFF	The type of XON/XOFF software flow control used to stop a receiver from being overrun with messages from a sender. The types are: • `Input` (use XON/XOFF to control the flow of data on the receive line) • `None` (default) • `Output` (use XON/XOFF to control the flow of data on the transmit line)

Property	Description
Handshaking Options	**RS-232** • **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-422** • No handshaking options are available for RS-422. **RS-485** • **Enable Stallion RS-485 Half Duplex**. Select if RS-232 to RS-485 is done using a Stallion EasyConnection adapter. • **Echo (Required for Stallion RS-485 ports)**. Select so that the server expects the messages it sends to the port on the transmit line to be echoed back on the receive line. Select for a Stallion EasyConnection adapter or a Black Box converter.

Terminal Server port properties

Property	Description
Terminal Server TCP Host Name Terminal Server TCP Port No	The name and port number of terminal server to which the channel is connected. You can specify either a TCP host name or an IP address, but it must match the TCP host name used when you installed and internally configured the terminal server.
Idle Timeout	The time, in seconds, the channel waits for a successful connection to the server before closing the connection. A value of *0* indicates that the connection is never closed.
Checksum	The type of checksum error detection used for the port. Not applicable for this channel. Select *NONE*.

Redundant port properties for a Honeywell Universal Modbus channel

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.

Main properties for a Honeywell Universal Modbus controller

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

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

Property	Description
Name	The unique name of the controller. A maximum of *10* alphanumeric characters (no spaces or double quotes). Note: In Station displays, underscore characters (_) appear as spaces. For LAN–connected controllers, the name is used to look up the IP address in the TCP/IP database if you do not specify an **IP Address** property. In the case of communications redundancy when the IP Addresses are not defined in Quick Builder, the IP Address 1 and 2 must be specified in the server hosts file. The host name for IP Address 1 is then the Name property with an 'A' appended to it and the host name for IP Address 2 is the Name property with a 'B' appended to it.
Description	(Optional) A description of the controller. A maximum of *132* alphanumeric characters, including spaces.
Channel Name	The name of the channel on which the controller communicates with the server. (You must have already defined a channel for it to appear in this list.)
Marginal Alarm Limit	The communications alarm marginal limit at which the controller is declared to be marginal. When this limit is reached, a high priority alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. A controller barometer monitors the total number of requests and the number of times the controller did not respond or response was incorrect. The barometer increments by two or more, depending on the error, and decrements for each good call. The default value is *25*.

Property	Description
Fail Alarm Limit	The communications alarm fail limit at which the controller is declared to have failed. When this barometer limit is reached, an urgent alarm is generated. To change the priority of the alarm system wide, see the topic titled "Configuring system alarm priorities" in the *Server and Client Configuration Guide*. To change the priority of the alarm for one controller, see the topic titled "About configuring custom system alarm priorities for an individual channel or controller" in the *Server and Client Configuration Guide*. Set this to double the value specified for the controller Marginal Alarm Limit. The default is *50*.
Dynamic Scanning Fastest Scan Period	Select the **Dynamic Scanning** check box to enable dynamic scanning of all point parameters on this controller. The default setting for this check box is selected. Define the fastest possible scan period (in seconds) that dynamic scanning will scan point parameters on this controller. The default is *15* seconds. The dynamic scanning period does not affect the static scanning rate for a parameter. For example, if the scanning rate for a parameter is 10 seconds, and the dynamic scanning rate for the controller is 15 seconds, the parameter will still be scanned at a period of 10 seconds.
Device Type	Select from the list, or enter the acronym for, the type of controller you are using. See the section below titled "Available device types." When the device type is HC900, the **Data Table** list appears.
IP Address	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. You can specify the port number to use. The ability to define a specific port enables multiple Modbus devices to be addressed behind a single IP address. If no port number is specified, port number 502 is used by default.
Device Identifier	The Universal Modbus identification number assigned to your device.
Data Table	Available for HC900 device types only. The data table used to address the controller. For a description of available data tables, see the section below titled "Data Table types."

Property	Description
Offset	Enter the lowest address within the range you intend to use. See the section below titled "Using offsets." By default use *0*. Note: **Offset** is not visible when **Device Type** is *HC900* and **Data Table** is *Holding register*.
Item Type	The type of controller specified when this item was created.
Last Modified	The date and time the controller properties were modified.
Last Downloaded	The date and time the controller was last downloaded to the server.
Item Number	The unique item number currently assigned to this controller, in the format *RTUnnnnn*. You can change the item number if you need to match your current server database configuration. The number must be between *01* and the maximum number of controllers allowed for your system. For more information about setting the maximum value, see the topic titled "Adjusting sizing of non-licensed items" in the *Supplementary Installation Tasks Guide*.

Data Table types

Data Table	Table Number	RTU Type	Address Range
All tables	0, 1, 3, 4	0	0x0–0x1fff (per table)
Holding Register	4	4	0x0–0x7fff

Property	Description
Data Table	The name of the data table.
Table Number	The table number to which the Data Table can access. See the section titled "Table types" in the topic titled "Address syntax for named addresses."
RTU Type	A number the system uses internally to identify the Data Table.
Address Range	The range of addresses the Data Table can access.

Available device types

Type Acronym	Controller Device
UDC2300	Universal Digital Controller 2300
UDC3300	Universal Digital Controller 3300

Type Acronym	Controller Device
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/X-Series Paperless Recorder

Using offsets

The server can only access a maximum of 4,096 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 Universal 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.

The exception to using offsets are those controllers with a **Data Table** setting of *Holding Register*. These controllers can access addresses between 0x0000 and 0x7fff without using an offset.

HC900 and UMC800 controller OFFSET addresses

Attention

This section is not applicable for controllers with a **Data Table** setting of *Holding Register*.

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 1,000 Signal Tags would require setup of two virtual controllers with offset entries of 0 and 2,000 respectively. HC900 control loops 25 through 32 parameters would require an offset entry of 6,000.

Parameter Category	OFFSET Address for Controller		Point Addressing
	UMC800	HC900	
Control Loops	0 (loops 1–16)	0 (loops 1–24)	Named (acronyms)
Control Loops (25–32)[36]	Not applicable	6,000	Named (acronyms)

Parameter Category	OFFSET Address for Controller		Point Addressing
	UMC800	HC900	
Variables (MATH_VAR)	0 (all Variables, 1–150)	0 (all Variables, 1–600)	Named (acronyms)
SP Programmers 1–4	0	0	Named (acronyms)
SP Programmers 5–8	Not applicable	Not supported	
Signal Tags (TAG)	2,000 (Signal Tags 1–500)	2,000 (Signal Tags 1–1,000)	Named (acronyms)
Signal Tags 1,001–2,000	Not applicable	4,000	Modbus Hexadecimal codes
SP Scheduler 1	2,000	2,000	Named (acronyms)
SP Scheduler 2	Not applicable	2,000	Named (acronyms)
Sequencers 1–4	Not applicable	4,000	Modbus Hexadecimal codes
Alternator, Stage, Ramp, HOA, Device Control	Not applicable	6,000	Modbus Hexadecimal codes

Related topics

“Address syntax for named addresses” on page 89

[36] For Loop 32, MODE is supported for Auto/Manual only.

Setting up time synchronization for UMC800 and HC900 controllers

Use the **Time Sync Value** field in the **Advanced** tab to set up time synchronization for UMC800 and HC900 controllers.

Property	Description
Time Sync Value	Add the value *0*, *1*, or *2* to set up time synchronization, where: *0* = No time synchronization (default) *1* = UMC800 Time Sync using 4× registers *1be0-1be5* *2* = HC900 Time Sync using 4× registers *1df0-1df5*

Honeywell Universal Modbus points reference

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

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

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.

Related topics

"Defining a Honeywell Universal Modbus address for a point parameter" on page 88

Defining a Honeywell Universal Modbus address for a point parameter

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

```
ControllerName Address
```

Part	Description
ControllerName	The name of the Universal Modbus controller.
Address	The address in the controller where the value is recorded. The syntax depends on the address type: • Named addresses • Non-named addresses

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

Related topics

“Planning considerations for installing and configuring Honeywell Universal Modbus controllers” on page 7

Address syntax for Honeywell Universal Modbus controllers

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

```
ControllerName Address
```

Part	Description
ControllerName	The name of the Universal Modbus controller.
Address	The address in the controller where the value is recorded. The syntax depends on the address type: • Named addresses. See the topic titled "Address syntax for named addresses" for more information. • Non-named addresses. See the topic titled "Address syntax for non-named addresses" for more information.

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

Related topics

“Address syntax for named addresses” on page 89

“Address syntax for non-named addresses” on page 93

Address 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
AddressName	Matches an address from the table of non-numbered addresses.
Format	(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
AddressName	Name of the address, for example, loop. For address names see the topic titled "Numbered addresses."
Number	The number of the address. For address numbers see the topic titled "Numbered addresses."
SubAddress Name	(Optional) Some types of numbered addresses can have sub-addresses. For example, every loop has a Process Variable (PV) and a Set Point (WSP).

Part	Description
Format	(Optional) Specify only if the device does not use the default format. See the topic titled "Data formats."

Examples

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 n PV*	Not configurable
Set Point (SP)	*LOOP n WSP*[37]	*LOOP n WSP*
Output (OP)	*LOOP n OPWORK*[38]	*LOOP n OPWORK*
MODE (MD)	*LOOP n LOOPSTAT*	*LOOP n MODEIN*

Loop tuning constants (possible AUX parameters for a loop point)

Parameter	Source Address	Destination Address
Gain	*LOOP n GAIN1*	*LOOP n GAIN1*
Reset	*LOOP n RESET1*	*LOOP n RESET1*
Rate	*LOOP n RATE1*	*LOOP n RATE1*

[37] LSP1 or SP1 can be used if the parameter WSP is unavailable.

[38] OP can be used if the parameter OPWORK is unavailable.

Digital Output values (used on a status point)

Attention

Digital Output values reside in table *0*. Therefore, they are not applicable for controllers with a **Data Table** setting of *Holding Register*.

Parameter	Source Address	Destination Address
PV	*DO n*	Not configurable
OP	*DO n*	*DO n*[39]

Digit Input values (used on a status point)

Attention

Digital Intput values reside in table *0*. Therefore, they are not applicable for controllers with a **Data Table** setting of *Holding Register*.

Parameter	Source Address	Destination Address
PV	*DI n*	Not configurable

Extended loop support for HC900

Attention

This section is not applicable for controllers with a **Data Table** setting of *Holding Register*.

Note that for LOOPX parameters, you must reference a controller with an Offset of 6,000. For more information about offsets, see the section "HC900 and UMC800 controller OFFSET addresses" in the topic titled "Main properties for a Honeywell Universal Modbus controller."

For the HC900 controller, loops 25-32 must be addressed using the LOOPX parameter. Loop 32 Mode is only supported for Auto/Manual (AUTO-LSP and MAN-LSP). Loop 32 addresses must be added via the custom Modbus map for full support of Mode.

[39] Note that not all devices support writing to digital output addresses. See the topic titled "Digital output control strategies."

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

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

Parameter	Source Address	Destination Address
Gain	*LOOPX n GAIN1*	*LOOPX n GAIN1*
Reset	*LOOPX n RESET1*	*LOOPX n RESET1*
Rate	*LOOPX n RATE1*	*LOOPX n 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 n*	Not Configurable

Digital signal tag example

Parameter	Source Address	Destination Address
PV	*TAG n*	Not Configurable

Digital variable example

Parameter	Source Address	Destination Address
PV	*MATH_VAR n*	Not Configurable
OP	*MATH_VAR n*	*MATH_VAR n*

Analog variable example

Parameter	Source Address	Destination Address
PV	*MATH_VAR n*	Not Configurable
SP	*MATH_VAR n*	*MATH_VAR n*

Related topics

“Main properties for a Honeywell Universal Modbus controller” on page 80
“Address syntax for Honeywell Universal Modbus controllers” on page 88
“Address syntax for non-named addresses” on page 93
“Numbered addresses” on page 13
“Data formats” on page 95
“Digital output control strategies” on page 95

Address syntax for non-named addresses

Addresses without names can be addressed directly using the format:

```
n:0xA [Format]
```

Part	Description
n	Table number. See the section below titled "Table types" for table descriptions and their number.
A	Address within the table (in hexadecimal).
Format	(Optional) Only used for Input and Holding register tables (3 and 4). If a format is not specified, the format defaults to IEEEFP.

For signal tags above 1,000, you must use hexadecimal addressing.

Table types

Attention

For controllers with a **Data Table** setting of `Holding Register`, you can only use Table 4.

Table Description	Table Number	Point Type	Address Type
Digital Output (also known as Coil)[40]	0	Status	Source/Destination[41]
Digital Input	1	Status	Source
Input Registers	3	Status/Analog/Accumulator	Source
Holding Registers	4	Status/Analog/Accumulator	Source/Destination

Non-named address examples (for HC900)

Parameter	Point Type	Address Type	Point Table/ Address/ Format	Controller OFFSET Address[42]
Signal Tag 1001	Analog	Source	4:0x4330 IEEEFP	4,000
Analog In, Slot 2, Channel 2, of Rack 2[43]	Analog	Source	3:0x112 IEEEFP	0
Digital In, Slot 8, Channel 3, of Rack 1[43]	Status	Source	1:0x72	0

[40] See the topic titled "Troubleshooting problems with specific controller models."

[41] See the topic titled "Digital output control strategies."

[42] **Offset is not required for controllers with a Data Table setting of `Holding Register`.**

[43] For controllers with a **Data Table** setting of `Holding Register`, DO and DI cannot be used. See the sections "Digital Output values" and "Digital Input values" in the topic titled "Address syntax for named addresses."

For controllers with a **Data Table** setting of `All Tables`, all I/O beyond Rack 1 in an HC900 must be addressed using hexadecimal 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 hexadecimal addresses.

Parameter	Point Type	Address Type	Point Table/ Address/ Format	Controller OFFSET Address[42]
Step Number of Sequencer 1	Analog	Source	4:0x5AA9 UINT2 (16-bit Integer)	4,000

Related topics

“Address syntax for Honeywell Universal Modbus controllers” on page 88

“Address syntax for named addresses” on page 89

“Troubleshooting problems with specific controller models” on page 125

“Digital output control strategies” on page 95

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 (0–15). 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

Related topics

“Address syntax for named addresses” on page 89

Digital output control strategies

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

[42] **Offset is not required for controllers with a Data Table setting of *Holding Register*.**

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 "Figure 12: Digital Output Control Strategy - Example 1", rather than use 'Digital Output' as the destination of a server point parameter, use the strategy shown in the "Figure 13: Digital Output Control Strategy - Example 2". 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`*.

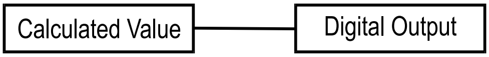

Figure 12: Digital Output Control Strategy - Example 1

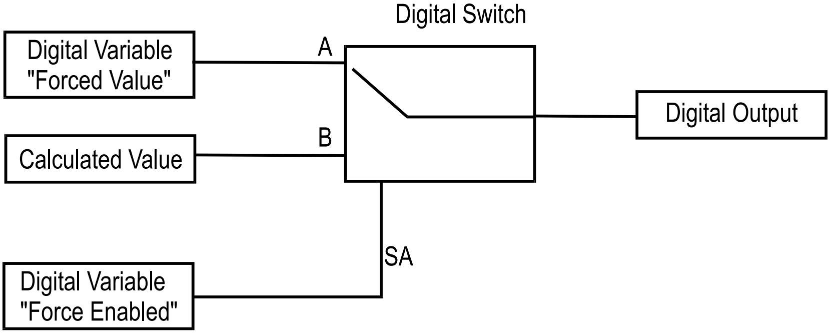
Digital Switch
Digital Variable "Forced Value"
A
B
Calculated Value
Digital Output
SA
Digital Variable "Force Enabled"

Figure 13: Digital Output Control Strategy - Example 2

Related topics

"Address syntax for named addresses" on page 89

"Address syntax for non-named addresses" on page 93

HC900 history backfill

Some versions of HC900 support historical data collection. This section describes how to configure and enable history backfill for supported HC900 controllers.

Related topics

“About history backfill” on page 98

“About configuring history backfill” on page 99

“Enabling history backfill” on page 100

“Operational aspects of history backfill” on page 101

“Troubleshooting history backfill” on page 102

About history backfill

HC900 firmware versions later than v4.4 support an historical data collection function. The Trend Point function blocks added to the controller configuration allow the controller to buffer point values that are collected at configurable rates.

When this function is configured in the controller, the buffered data can be used to populate history on the Experion server if a controller is disconnected from the server and cannot collect point values via normal scanning. The process used is called history backfill and is initiated when the controller comes back online after it has been disconnected for more than one minute. This disconnection could be due to:

- The controller failing due to a communications failure
- The controller being disabled (Out of Service)

The history backfill function only supports standard history 1 minute snapshots for analog and status points. When history is backfilled, value/timestamp pairs are read from all configured trend buffers in the controller. These are then inserted into standard history snapshots as if they were read by the server at the value's timestamp.

CAUTION

For points that have history backfill configured:

- Standard history averages are NOT re-averaged
- Any Status point PV or OP with non-consecutive addressed bits is not recalculated

The history backfill function is a licensable option.

About configuring history backfill

Most of the configuration tasks associated with using the HC900 history backfill functionality must be performed at the controller level using the HC (Hybrid Control) Designer software. This involves configuring a Trend Rate function block and several Trend Point function blocks on the controller. Refer to the HC (Hybrid Control) Designer documentation for further details.

After you have assigned points to the Trend Point function block in the controller, these points must also be assigned to standard history on the server. This can be done using Quick Builder (recommended) or directly in Station via the History Assignment displays. You can optionally configure History Archiving (see the topic "History Archiving" in the *Server and Client Configuration Guide*).

Configuring history backfill

To allow the history backfill process to operate, you need to change some server settings. Use Station to make these changes.

Attention

Selecting to enable history backfill using the steps below does not trigger a backfill of history; it only enables the history backfill option.

To trigger a backfill of history for a particular controller, go to that controller's status display and clear the **Enable History Backfill** check box for at least one minute. Then, select the **Enable History Backfill** check box. This will trigger history backfill for that controller.

To configure history backfill:

1. From the Station menu, choose **Configure** > **System Hardware** > **Server Wide Settings**.
2. Select the **History Backfill** tab.
3. Select the **Enable history backfill** check box to enable history backfill.
4. In the **Maximum number of days to backfill** box, enter the maximum number of days that history can be backfilled (23 recommended).

   Any event data older than this number of days will be discarded and will not be inserted into history. This maximum number of days must not exceed the number of days of standard history snapshots that is configured for the system.

Enabling history backfill

Prerequisites

- The HC900 controller has been configured with trend points.

To enable history backfill for HC900

1. In Station, choose **Configure** > **System Hardware** > **SCADA Controllers**.
2. Click the HC900 controller.
   The Controller Status display appears.
3. Under History Backfill, select the **Enable** check box.

Operational aspects of history backfill

CAUTION

For points that have history backfill configured:

- Standard history averages are NOT re-averaged
- Any Status point PV or OP with non-consecutive addressed bits is not recalculated

A history backfill will be initiated for a controller when any of the following events occur:

- If the controller has been in the 'Failed' or 'Marginal' state for more than 1 minute and then returns to the OK state. History backfill needs to be Enabled as well.
- If the controller has been Disabled (Out of Service) for more than 1 minute and is then Enabled (In Service) and returns to the OK state. History backfill needs to be Enabled as well.
- If the controller is in the OK state and history backfill has been disabled for more than 1 minute and is then re-enabled.

Attention

History samples will only be backfilled if they have not been collected by the Server by normal scanning.

Timestamps

If a Trend Rate function block has been configured in the controller, up to three buffers are allocated in the controller to store the data that is collected. Each of these buffers has two timestamps; one to indicate the oldest sample it stores and another to indicate the most recent sample it stores.

These timestamps can be viewed on the Controller Status display for the HC900 controller. These timestamps are only updated when the controller is enabled (In Service).

Alarms and events

When a history backfill occurs an event is raised to indicate the controller that was backfilled. If any missed samples are detected (that is, neither the controller nor the server collected the samples) an alarm is raised indicating the controller, the point and the time period when samples were missed.

Troubleshooting history backfill

The controller performs a history backfill and there are trend gaps/missed samples even though there has been no problem with the controller.

Diagnostic check

Using the HC Historian application, ensure that the controller has missed the specified samples.

Cause

A possible cause could be that the controller time is being changed or is being synchronized from another time source and the time has been moved forward.

Solution

Ensure only one server is connected to this controller. Also ensure that an engineering tool, such as HC Designer, has not changed the controller time.

HC900 and UMC800 SPP and recipe support

This section describes set point programmers and recipe support for HC900 and UMC800.

Related topics

Overview of SPP and Recipe Support

The HC900 and UMC800 SPP and 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').

Attention

Set point programs and recipe management are not available from Console Stations.

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.

Related topics

"Resource requirements" on page 105

"Migration conversion requirements" on page 106

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 1,000 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 1,000 server database profiles.

The server database also allows you to configure and store up to 1,000 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 1,000 server database recipes.

Set point program history

The history of an 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 1,000 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 1,000 profiles.

The utility is named *umc800export*, and may be run from the command line.

To migrate profiles

1. From a command line, type **`C:> umc800export`** and then press Enter.

   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 1,000 recipes. Recipes need to be recreated manually in Station.

Recipe 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.

Related topics

“Configuring a recipe” on page 107

“Configuring an SP profile” on page 109

“Configuring a combined recipe” on page 110

“Configuring SPP monitoring” on page 112

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 1,000 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** drop-down list, the server attempts to read a new list of variables from the controller. If the controller is not a valid HC900 or UMC800 controller or the upload fails, an alarm is raised.

Attention

Only the first 75 variables configured on an HC900 are used to populate the variable list. While additional variables (up to 255) are not visible in the list, they can still be added to a recipe.

HC900 controllers prior to Rev 4.0 firmware support 8–character variable names. HC900 controllers Rev 4.0 and above support 16–character variable names.

The recipe download will fail if the variable name does not exist in the controller or a variable name is repeated in the 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 topic titled "Downloading a recipe" for information on downloading a recipe.

See the topic titled "Configuring a combined recipe" for information on configuring a recipe for use in a combined recipe.

Related topics

“Downloading a recipe” on page 115

“Configuring a combined recipe” on page 110

Configuring an SP profile

An 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 a soak, except the last segment, which 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 1,000 profiles can be created and maintained using the Station HC900 and UMC800 Profile Configuration displays.

To configure an SP profile

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 an 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 1,000 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 or changes to program segments to an HC900 SP programmer, provided the programmer is in the Ready or Hold mode. See the section 'Downloading an SP profile' for information on downloading a profile.

Related topics

"Configuring a combined recipe" on page 110

"Downloading an SP profile" on page 116

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 1,000 combined recipes can be created and maintained using the Station HC900/ UMC800 Combined Recipe Configuration displays.

To configure a combined recipe

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 1,000 recipes stored in the server database (see the topic titled "Configuring a recipe" for information on recipes). The remaining components are up to two SP profiles, selected from the 1,000 profiles stored in the server database (see the topic titled "Configuring an SP profile" 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 20 *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 topic titled "Downloading a combined recipe" for information on downloading a combined recipe.

Related topics

“Configuring an SP profile” on page 109

“Downloading a combined recipe” on page 117

“Configuring a recipe” on page 107

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 an SP programmer, and does not store the program in the server database.

The SPP Trend display allows the user to view the history of an 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.

Parameter	Source Address
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 its A1 parameter to read the calculated PV from CARB5.

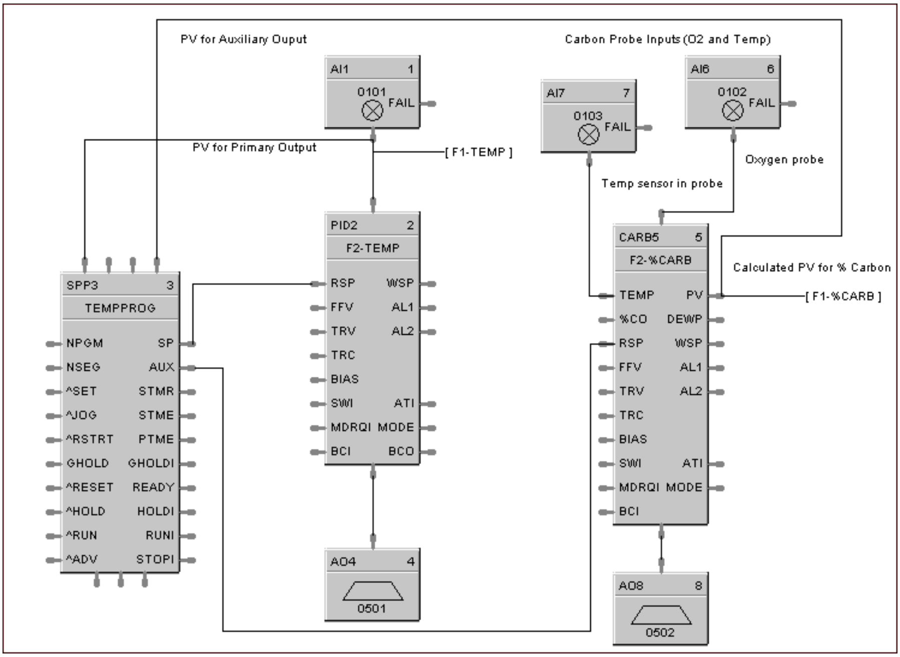

Figure 14: Example SPP Implementation

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 *Server and Client Configuration Guide* for information on points.

To configure SPP monitoring

1. Disable the HC900 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 and 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.

Related topics

"Downloading a recipe" on page 115

"Downloading an SP profile" on page 116

"Downloading a combined recipe" on page 117

"Controlling an SP programmer" on page 117

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 **Download to Controller** 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 topic titled 'Troubleshooting recipe support' for possible fail reasons.

Related topics

"Configuring a recipe" on page 107

"Troubleshooting recipe support" on page 119

Downloading an SP profile

CAUTION

A profile can only be downloaded while the selected programmer is not running. Make sure that the programmer is in the Ready mode.

To download an 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. Check that the programmer to be selected is in the **Ready** mode. Downloads will not be accepted while the programmer is running.
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 topic titled 'Troubleshooting recipe support' for possible fail reasons.

Related topics

"Configuring an SP profile" on page 109

"Troubleshooting recipe support" on page 119

Downloading a combined recipe

CAUTION

If the download includes a recipe, then running programs can be affected by changing the variable values. The SP programmers must be in **Ready** or **Hold** state to download the profiles.

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. Select the combined recipe name to load its configuration.
4. Click **Download** 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 topic titled 'Troubleshooting recipe support' for possible fail reasons.

Related topics

"Configuring a combined recipe" on page 110

"Troubleshooting recipe support" on page 119

Controlling an SP programmer

Considerations:

- You can change the present segments 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 an SP programmer

1. In Station select **Configure** > **Applications** > **HC900/UMC800** > **Programmer Operation**.

   The SPP Summary display opens.
2. Select a controller in the combo box.

   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 topic titled 'Troubleshooting recipe support' 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, an 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 restarted.

Related topics

“Troubleshooting recipe support” on page 119

Troubleshooting recipe support

This section describes cross checks and remedies to perform if HC900/UMC800 SPP and 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 an 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' or 'Hold' 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 an 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.

Related topics

“Controlling an SP programmer” on page 117

“Downloading a combined recipe” on page 117

“Downloading a recipe” on page 115

“Downloading an SP profile” on page 116

Troubleshooting Honeywell Universal Modbus issues

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

Related topics

"Testing Honeywell Universal Modbus communications with the server" on page 122
"Troubleshooting Honeywell Universal Modbus configuration errors" on page 124

Testing Honeywell Universal Modbus 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.

Example

```
C:\>umbtst

Enter LRN or device name of channel
chn01
Enter Controller number
1
Enter command:
?
READ i,a,n,p - read device i address a for n address
and p passes
WRITE i,a,d,p - write device i address a data d for p passes
DUMP i,a,b - read device i address a to address b
FIND i,j - find device with id i to j
DELAY n - set delay between passes to n ms
FORMAT f - display registers in format f (DEC,HEX,or IEEEFP)
FO filename - direct output to file
! - execute last command
Q - quit
Enter command:
find 1,4

FIND device with id   1 to   4, at 28-May-12  14:06:52
```

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

The *FIND* command is used with serial connected devices to check if the devices are present. This command locates all Universal Modbus devices on the channel with IDs between *a* and *b*. This command is not applicable on LAN-connected devices.

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 Universal Modbus channel *COM3* might be channel number *1*. Its device name will be *chn01*.

Related topics

"Planning considerations for installing and configuring Honeywell Universal Modbus controllers" on page 7

Troubleshooting Honeywell Universal Modbus configuration errors

Related topics

"Troubleshooting Honeywell Universal Modbus common problems" on page 124

"Troubleshooting problems with specific controller models" on page 125

"Planning considerations for installing and configuring Honeywell Universal Modbus controllers" on page 7

Troubleshooting Honeywell Universal Modbus common problems

Error message or problem	Description
You see the following error in the server log file: `Error code 0106 (Device Timeout)`	The server has not received a response from the controller.
You see the following 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 following 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).

Error message or problem	Description
You see the following 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 *n* 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.

Troubleshooting problems with specific controller models

UDC3300 problems

Error message or problem	Description
• You attempt to change a set point in Station and the value changes on the Station display but not on the controller faceplate. • You attempt to change a set point 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 or problem	Description
• You know that you should use the address `LOOP n parametername` but you don't know what value to use for *n*. • You want to access the process variable of the only PID loop you have configured. You used the number *n* which appears on the top right-hand corner of the PID block (`LOOP n 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 n PV` doesn't download.	The PID part of the address is not valid and doesn't appear in the Universal Modbus driver documentation. You cannot use the names of control blocks within Control Builder as Universal Modbus addresses. You can only use the addresses listed in the Universal Modbus documentation.
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.

Error message or problem	Description
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. Refer to the Universal Modbus documentation for information 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. Refer to the Universal Modbus documentation for information about read-only and write-only addresses.
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**, and then click **OK**. 4. Your printout should show 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 n` or `MATH_VAR n`.

HC900 problems

Error message or problem	Description
You know you should use the address LOOP *n parametername* but you don't know what value to use for *n*. You want to access the process variable of the only control loop that you have configured. You used the number *n* which appears on the upper-right hand corner of the PID block (LOOP n PV for loops 1-24 and LOOPX n 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 the FBD 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 *n* but you don't know what value to use for *n*.	For the first rack only, the analog input number is calculated using the formula: $n = (m\text{-}1) * 8 + c$. 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 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 or problem	Description
I don't know to access analog inputs beyond the first rack.	A Signal Tag is the only way to access analog inputs on a Holding Register controller. Access to analog inputs directly using Non-Named hexadecimal addressing requires Table 3 (Input Register), whereas Signal Tags use Table 4 (Holding Register). 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 IEEEFP, channel 2 is 3:x102 IEEEFP, channel 8 is 3:x10E IEEEFP. 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 or problem	Description
I don't know to access digital I/O beyond the first rack.	A Signal Tag is the only way to access digital inputs and outputs on a Holding Register controller. Access to digital inputs and outputs directly using Non-Named hexadecimal addressing requires Table 1 (Digital Input) and Table 0 (Digital Output), whereas Signal Tags use Table 4 (Holding Register). 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.

Error message or problem	Description
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 an `All Tables` controller whose **OFFSET** address is not 2,000. Refer to the Universal Modbus documentation about offset 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. You will need to use Variables in your HC900 configuration instead for writes. Refer to the Universal Modbus documentation for information 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 **Modbus Register Map** > **Signal Tags and Variables** 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 *n* (for Signal Tags) or MATH_VAR *n* 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.

Error message or problem	Description
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).
You see the error: `***** PNTBLD ERROR ***** Table Number must match controller Table Number (Table Number: x, controller Table Number: 4)`	This error occurs when you used a Holding Register controller ("controller Table Number: 4"), but you specified an address that resides in a different table ("Table Number: *x*." – *x* could be 0, 1, or 3). A Holding Register controller only supports addresses in the Holding Register table. Most named addresses reside in the Holding Register table, but some, such as DI and DO, reside in different tables. If you need to access Digital Outputs, Digital Inputs, or Analog Inputs (Input Registers) on a Holding Register controller, you can connect these to Signal Tags or Analog/Digital Variables, and then use the `TAG` and `MATH_VAR` named addresses.
You see the error: `*** Cannot change Data Table - set Data Table back to All tables (RTU type: 0)`	Once a controller has been downloaded to the server, its Data Table value cannot change. This message informs you of this, and notifies you of the accepted (or previously downloaded) value for this particular controller ("All tables").

Related topics

"Address syntax for non-named addresses" on page 93

Notices

Trademarks

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

OneWireless™ is a trademark of Honeywell International, Inc.

Other trademarks

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

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

Third-party licenses

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

Documentation feedback

You can find the most up-to-date documents on the Honeywell Process Solutions support website at:

http://www.honeywellprocess.com/support

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

hpsdocs@honeywell.com

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

How to report a security vulnerability

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

Honeywell investigates all reports of security vulnerabilities affecting Honeywell products and services.

To report a potential security vulnerability against any Honeywell product, please follow the instructions at:

https://honeywell.com/pages/vulnerabilityreporting.aspx

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

- Send an email to security@honeywell.com.

  or

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

Support

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

Training classes

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

Index