Honeywell

HC900 Hybrid Controller Communications User Guide

51-52-25-111

1/06

Revision: 6

Copyright, Notices, and Trademarks

Revision 6 – 1/06

Warranty/Remedy

Honeywell warrants goods of its manufacture as being free of defective materials and faulty workmanship. Contact your local sales office for warranty information. If warranted goods are returned to Honeywell during the period of coverage, Honeywell will repair or replace without charge those items it finds defective. The foregoing is Buyer's sole remedy and is **in lieu of all other warranties, expressed or implied, including those of merchantability and fitness for a particular purpose**. Specifications may change without notice. The information we supply is believed to be accurate and reliable as of this printing. However, we assume no responsibility for its use.

While we provide application assistance personally, through our literature and the Honeywell web site, it is up to the customer to determine the suitability of the product in the application.

Honeywell
Industrial Measurement and Control
1100 Virginia Drive
Fort Washington, PA 19034

Reference: Modicon Modbus Protocol Reference Guide - PI-MBUS-300 Rev. G

About This Document

Abstract

This document provides information specific to the communications interface for Honeywell's HC900 Controller. The protocol supported for connection to the controller's Ethernet network port is Modbus/TCP (Modbus RTU protocol in a TCP/IP wrapper). . The document includes a summary of all HC900 data available (primarily floating point) for Modbus RTU access read and write including methods for access.

Contacts

World Wide Web

The following lists Honeywell's World Wide Web sites that will be of interest to our customers.

Honeywell Organization	WWW Address (URL)
Corporate	*http://www.honeywell.com*
Industrial Measurement and Control	*http://www.honeywell.com/imc*

Telephone

Contact us by telephone at the numbers listed below.

	Organization	Phone Number	
United States and Canada	Honeywell	1-800-423-9883 1-888-423-9883 1-800-525-7439	Tech. Support Q&A Faxback (TACFACS) Service

Contents

Tables

Figures

1. Introduction

1.1 Overview

The HC900 controller provides Modbus communication support on three communication interfaces.

- Network port: Modbus TCP on an Ethernet connection
- Serial Port S1 RS232/RS485 selectable port. (Default RS232.): Modbus RTU
- Serial Port S2 RS232/RS485 selectable port. (Default RS485.): Modbus RTU

The user may find it convenient to print out the Modbus addresses for various parameters of the configuration (signal tags, PID loops, SP programmer, etc.) using the HC Designer report functions. With HC Designer Ver. 2.1 and later, these reports may also be exported to .csv files for view/manipulation in a spreadsheet and possible import to other HMI applications.

1.2 Modbus/TCP Interface

Introduction

HC900 controllers support the Modbus/TCP (also called Modbus TCP/IP or Modbus Ethernet) protocol for communications with third party HMI and SCADA software via a direct Ethernet TCP/IP connection.

The controller's Ethernet 10/100Base-T Host port is used for the Modbus/TCP connection. Ethernet TCP allows multiple concurrent connections to hosts for data interchange. The HC900 (C30/C50) supports 5 concurrent host connections using Modbus/TCP protocol messaging via this port; HC900 (C70/C70R) supports 10.

Interface Preparation

ATTENTION

To access the controller you must have a current Hybrid Control Designer configuration file available. Some data is referenced relative to number, such as Signal Tags and Variables.

Other principal blocks, such as PID blocks, have offsets for parameter access dependent on the order in which the blocks were placed on the Function Block Diagram.

It is strongly recommended that you upload the controller configuration using the Hybrid Control Designer configuration tool to assure that you have a current configuration.

The Hybrid Control Designer tool provides a series of reports for use in Modbus Address identification.
The **"Tag Information"** report lists the variables and Signal Tags in numeric order along with their Modbus Addresses.
A **"Block Modbus Address"** report lists the starting addresses for all principal blocks configured, identifying the offset.

Modbus/TCP Protocol

Modbus/TCP protocol, developed by Groupe Schneider's Modicon Division, is a popular, open standard for data interchange over Ethernet TCP/IP networks using a Modbus RTU command structure.

It is simply an encapsulation of Modicon's Modbus RTU protocol within a TCP/IP frame as shown below, which includes header information and the Modbus frame.

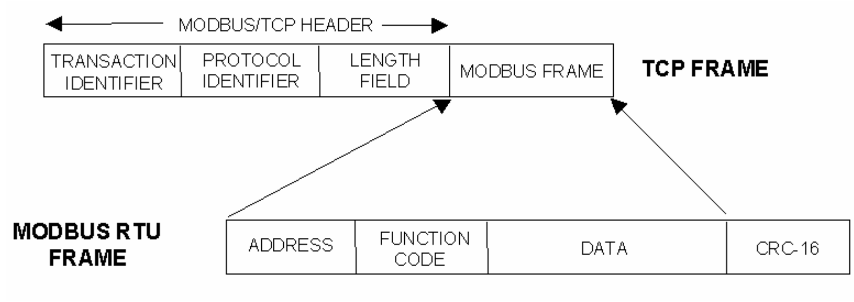


Figure 1-1 Modbus RTU Protocol within a TCP/IP Frame

The Open Modbus/TCP Specification is followed with respect to the physical, data link, and network layers. The message structure within the Modbus frame uses standard Modbus RTU function codes.

The Address part of the Modbus frame is not used (set to 00) since there is no sub-addressing intended or required. The controller IP address is the identifying address, set independently at the controller.

The error checking is supported by TCP/IP network protocols and not part of the Modbus frame.

The Transaction Identifiers and Protocol Identifiers in the header are normally all 0's (4 bytes total) while the Length field identifies the number of bytes in the Modbus frame. The controller will transmit the correct number of bytes for the remainder of the frame. However, the controller does not check this field for messages received.

The standard IEEE 32-bit floating point and 16-bit integer formats are used.

Parameter Addressing

The definition in Table 6-1 is the register map overview listing starting and ending addresses.

Greater detail for parameter addressing relating to a particular function class, e.g, loops, setpoint programmer, signal tags, etc. is in referenced sub-sections.   Function Codes 1, 2, 3, 4, 5, 6, 8, 16 (10h), and 17 (11h) are supported (see Table 4-1 Modbus/TCP and Modbus RTU Function Codes Definitions).

Examples for read or write access to parameters supported by the various function codes are provided in Sections 4.2 through 4.10.

Reference

The Open Modbus/TCP Specification can be obtained at the Modicon website:
http://www.modicon.com/openmbus/standards/openmbus.htm

HC900 Ethernet Communications Setup

See the HC900 Hybrid Control (HC) Designer Users Guide, Doc. # 51-52-25-110 or respective HC Designer Help Files for setting up the following network parameters:

IP Address, Subnet Mask (optional), Default Gateway IP Address (optional)

1. Be sure the PC, HMI panel, or other Host device has a Network Interface Card (NIC) with an IP address (fixed or DHCP served) that allows access to controllers on the same or other subnet. Consult your IT department or network administrator for allocating IP addresses to the controllers as required.
2. You will need to set each controller's IP address prior to network connection since every HC900 controller is shipped with the default IP address of 192.168.1.254. Placing multiple controllers on the same network before they have been given unique IP addresses will cause problems.
3. On the PC, use the Utilities Worksheet in the HC Designer software to set up the serial RS-232 connection to the controller at the desired baud rate. This will require a null modem cable.
4. Select the Set Controller's Network Parameters button. Using the wizard (bottom radio button), select the PC COM port to be used, then set the controller's new network parameters including IP address,

Subnet Mask (if other than default, 255.255.255.0), and Default Gateway IP address (if required, otherwise leave at default 0.0.0.0).   See your IT network administrator for proper entries. (Refer to the on-line help provided with the HC Designer software, **Utilities Worksheet, Set Controller's Network Parameters**, for further details on this step).

Note: This setup will require the controller to be placed temporarily in the Program mode. After the new network parameters have been downloaded, the controller will conduct a Cold Start in its transition to RUN. This will cause an initialization if there is a current configuration in the controller.

The fixed IP address of each controller shall be set independently prior to placing on the network. See your IT systems administrator for allocating IP addresses, subnet masks, or default gateway IP address as necessary (network address filtering and routing may be necessary if the controller network access will not to be confined locally within the plant environment).

Ethernet 10/100Base-T Network Connections

Ethernet 10/100Base-T networks operating at 10/100MB/sec. are supported. A typical network arrangement is as shown below.

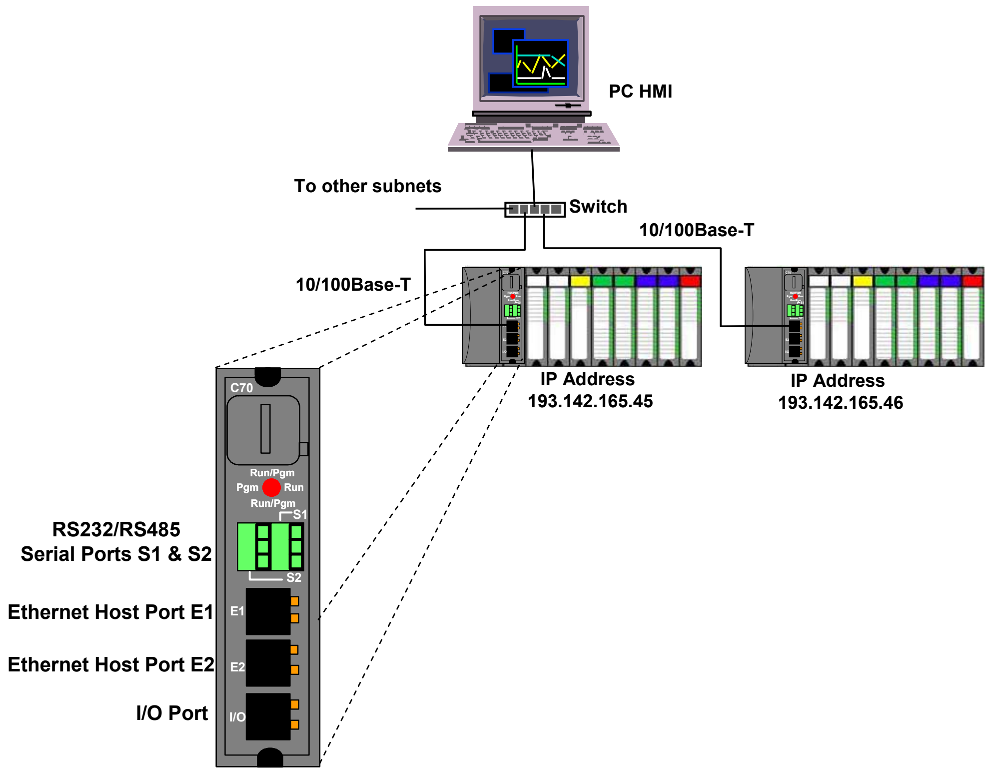


Figure 1-2 Ethernet 10/100Base-T Network Connections

Setting Up the Modbus/TCP Double Register Format

The HC900 predominantly uses an IEEE floating point format for communicating data to software applications providing Modbus/TCP protocol communications drivers. A floating point value is sent as (2) consecutive 16-bit registers, each register of which consists of two 8-bit bytes. Some software packages require the registers and bytes to be sent in a certain order. The controller can be configured to deliver the data in four different byte orders.

The Hybrid Control Designer software tool allows this order to be selected as follows:

1. Using the Utilities Worksheet in the HC Designer software, access the Set Controller Network Parameters button and make the selection to change the Modbus TCP Double Register Format (middle radio button).
2. With the Port selected for downloading this order (using a COM port or Network port), select the appropriate byte order format if the default (FP B) is not appropriate for the application. See Table 3-1, page 10.
3. Select Next and verify (by the response in the dialog box) that the change has been made in the controller. This order can be changed in the RUN mode.

The Modbus TCP double register transmission format selection, FP LB "Little Endian Byte-Swapped", would be selected for interface to most third party software packages which use this format as standard. The default, FP B "Big Endian" is used with SpecView32 or Honeywell's PlantScape/Experion/EBI software and follows the "Honeywell" default format of other control and recording products. It should be noted that most PC software packages offer a register (word) swap selection in their driver package anyway, so there should never be an incompatibility.

1.3 Modbus RTU RS232/RS485 Communication Ports

This implementation is designed to provide a popular data exchange format connecting these instruments to both Honeywell and foreign master devices via the RS232 and RS485 communication ports. The Modbus RTU allows the instrument to be a citizen on a data link shared with other devices, which subscribe to the Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. G specification.

These instruments DO NOT emulate any MODICON type device. The Modbus RTU specification is respected in the physical and data link layers. The message structure of the Modbus RTU function codes is employed and standard IEEE 32-bit floating point and integer formats are used. Data register mapping is unique to these instruments. The definition in Table 6-1 is the register mapping for the HC900 and the corresponding parameter value within those instruments.

Modbus RTU Message Format

Table 1-1 Modbus RTU Message Formats

Coding system	8 bit binary
Number of data bits per character	10, 11, or 12 Bits start bits - 1 data bits - 8 parity bits – 0 or 1 selectable stop bits – 1 or 2 selectable
Parity	None, odd, even selectable
Bit transfer rate	9600, 19200, 38400, 57600 Selectable
Duplex	Half duplex Transceiver or TX/RX
Error checking	CRC (cyclic redundancy check)
Polynomial	(CRC-16 10100000000001)
Bit transfer order	LSB first
End of message	Idle line for 3.5 or more characters (>1.82 msec for 19200).

Modbus RTU Link Layer

The link layer includes the following properties/behaviors:

- Slave address recognition,
- Start / End of Frame detection,
- CRC-16 generation / checking,
- Transmit / receive message time-out,
- Buffer overflow detection,
- Framing error detection,
- Idle line detection.

Errors detected by the physical layer in messages received by the slave are ignored and the physical layer automatically restarts by initiating a new receive on the next idle line detection.

General Modbus RTU message format

Query message format
[Slave Address, Function Code, Function code dependent data, CRC 16]

Response message format
[Slave Address, Function Code*, Function code dependent data, CRC 16]

* If an error is detected in a valid message the response function code is modified by adding 80 (hex) and the function code dependent data is replaced by an exception response code as described in 5. Modbus RTU Exception Codes .

Between messages, the RS-485 link is in a high impedance state. During this time receiving devices are more susceptible to noise generated false start of messages. Although noise-generated messages are rejected due to address, framing, and CRC checking, they can cause the loss of a good message when they are included in the message stream. In the slave the transmitting device enables its transmitter line diver and forces an idle line state onto the link for three character time slots prior to transmitting. This forces termination of any noise generated messages and improves message frame synchronization.

Modbus RTU Data Layer

The data layer includes:

- Diagnostic loopback,
- Function code recognition / rejection,
- Busy / repoll,
- Data error code generation

Errors detected by the data layer are rejected and the slave responds to the polling device with a Modbus-type status exception error. A summary of the Modbus status exception codes is listed in Section 5. Modbus RTU Exception Codes.

2. IEEE 32-bit Floating Point Register Information

The Modbus interface supports IEEE 32-bit floating point information for several of the function codes.

2.1 IEEE Floating Point Data Format

The formula for calculating the floating point number is:

$$\text{mantissa x } 2^{(\text{exponent } -127)}$$

(23 bit signed binary with 8 bit biased binary exponent)

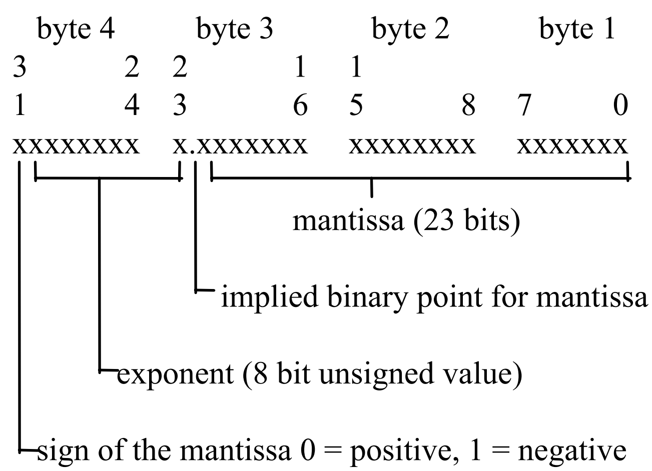


Figure 2-1 IEEE Floating Point Data format

Mantissa and Sign

The mantissa is defined by a sign bit (31) and a 23-bit binary fraction. This binary fraction is combined with an “implied” value of 1 to create a mantissa value, which is greater than or equal to 1.0 and less than 2.0.

The mantissa is positive if the sign bit is zero (reset), and negative if the sign bit is one (set). For example:

DECIMAL	HEXADECIMAL	BINARY
100	42C80000	01000010 11001000 00000000 00000000

The sign bit (31) is zero, indicating a positive mantissa. Removing the sign bits and exponent bits, the mantissa becomes:

HEXADECIMAL	BINARY
480000	xxxxxxxx x1001000 00000000 00000000

Add an “implied” value of one to the left of the binary point:

BINARY
1.1001000 00000000 00000000

Using positioned notation, this binary number is equal to:

$$1.0 + (1\text{x}2^{-1}) + (0\text{x}2^{-2}) + (0\text{x}2^{-3}) + (1\text{x}2^{-4}) = 1.0 + 0.5 + 0.0 + 0.0 + 0.0625 = 1.5625$$

Exponent

The exponent is defined by an unsigned 8-bit binary value (bits 23 through 30). The value of the exponent is derived by performing a signed subtraction of 127 (decimal) from the 8-bit exponent value.

DECIMAL	HEXADECIMAL	BINARY
100	42C80000	01000010 11001000 00000000 00000000

Removing the sign and mantissa bits, the exponent becomes:

DECIMAL	HEXADECIMAL	BINARY
133	85	x1000010 1xxxxxxx xxxxxxxx xxxxxxxx

or:

$$1x2^7 + 0x2^6 + 0x2^5 + 0x2^4 + 0x2^3 + 1x2^2 + 0x2^1 + 1x2^0$$

Subtract a bias of 127 (decimal) from the exponent to determine its value: 133 – 127 = 6.

Mantissa and Exponent Combination

Combining the mantissa and exponent from the two previous examples:

$$\text{float number} = \text{mantissa x } 2^{\text{exponent}}$$

$$\text{float number} = 1.5625 \text{ x } 2^6 = 1.5625 \text{ x } 64 = 100.0$$

Below is a list of sample float values in IEEE format:

DECIMAL	HEXADECIMAL
100.0	42C80000
-100.0	C2C80000
0.5	3F000000
-1.75	BFE00000
0.0625	3D800000
1	3F800000
0	00000000

Reserved Operands

Per the Standard certain exceptional forms of floating point operands are excluded from the numbering system. These are as follows:

EXCEPTION	EXPONENT	MANTISSA
+/- Infinity	All 1's	All 0's
Not-a-Number (NAN)	All 1's	Other than 0's
Denormalized Number	All 0's	Other than 0's
Zero	All 0's	All 0's

3. Modbus Double Register Format

Data that is 32 bits requires 2 sequential registers (4 bytes) to transfer its data. Data of this type includes IEEE 32-bit floating point, 32-bit signed integer and 32-bit unsigned integer. The stuffing order of the bytes into the two registers differs among Modbus/TCP hosts. To provide compatibility, the double register format for the HC900 controller is configurable.

To set the controller's double register byte order, go to the "Set Controller Network Parameters " wizard in the "Controller Utilities Function" section of the Utilities Tab on the Hybrid Control Designer and configure "Modbus Double Register Format". This can be done in the RUN mode.

The selections are:

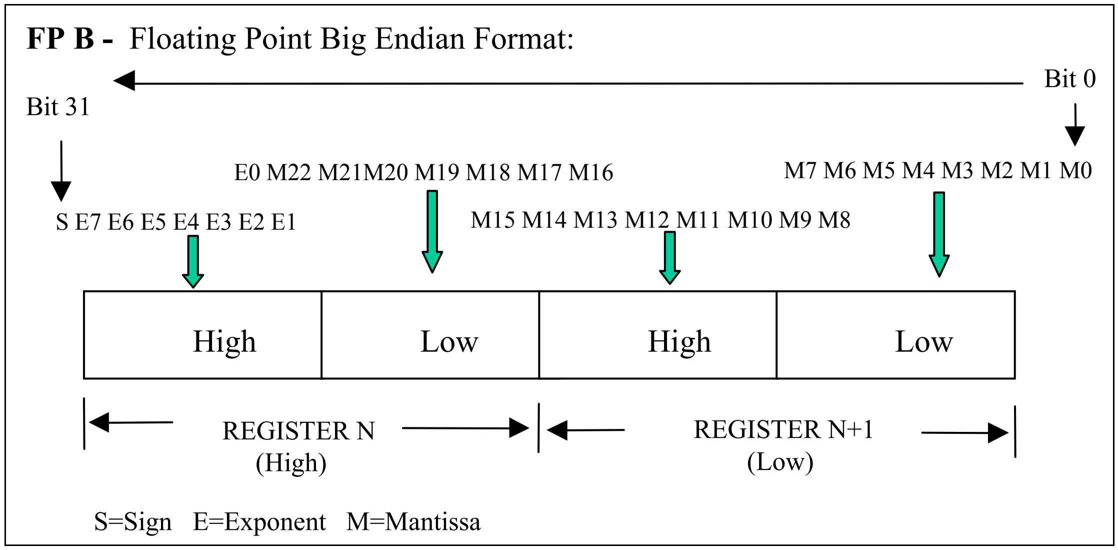
Table 3-1 Modbus Double Register Format Selections

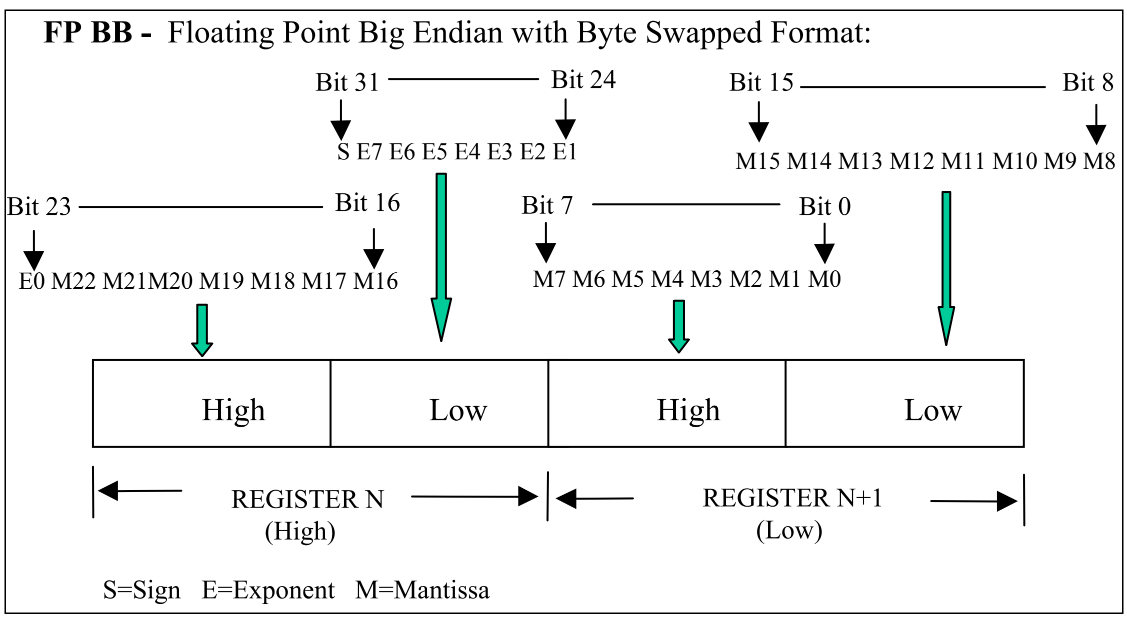
Selection	Description	Byte order (See Figure 2-1)	Notes
FP B	Floating Point Big Endian Format	4, 3, 2, 1	HC900 default
FP BB	Floating Point Big Endian with byte-swapped	3, 4, 1, 2	
FP L	Floating Point Little Endian Format	1, 2, 3, 4	
FP LB	Floating Point Little Endian with byte-swapped	2, 1, 4, 3	Modicon and Wonderware standard

See IEEE Formats on page IEEE Floating Point Formats on page 11 and 32-bit integer formats on page 13.

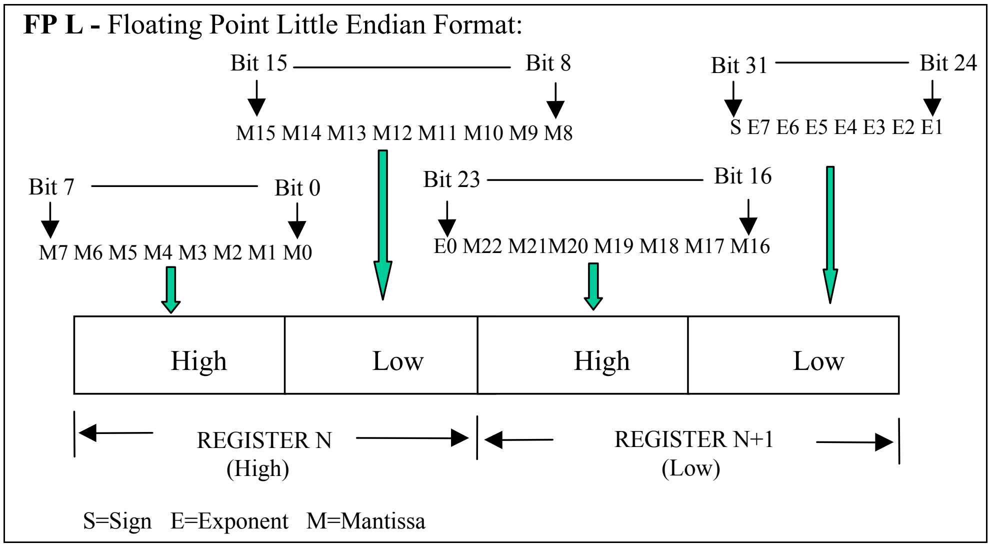
NOTE: Byte Swapping only applies to Function Codes 3, 4, and 16.

3.1 IEEE Floating Point Formats





continued next page



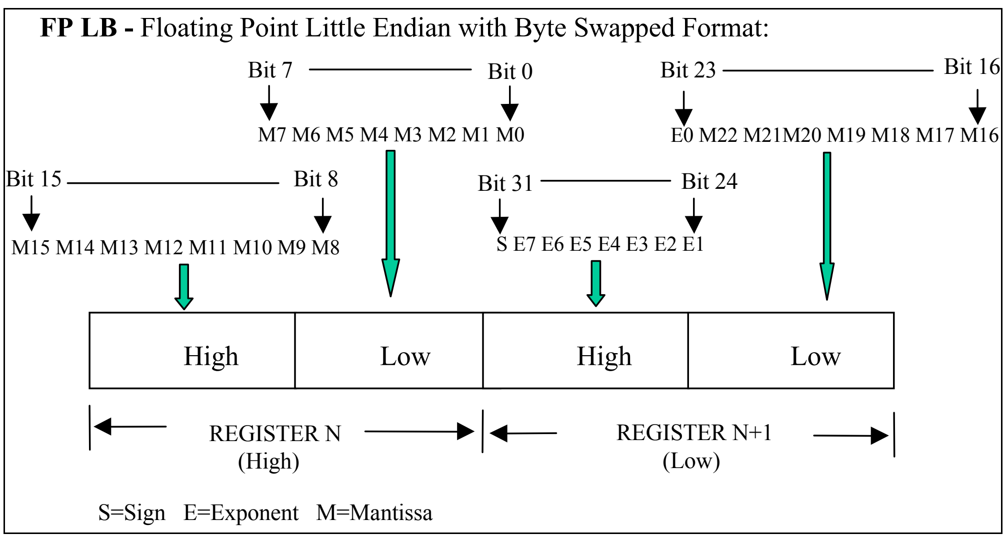


Figure 3-1 IEEE Floating Point Formats

Table 3-2 IEEE Floating Point Number Examples in FP B Format

Value (decimal)	IEEE FP B	Register N		Register N+1	
	MSB LSB	high	low	high	low
100.0	42C80000h	42h	C8h	00h	00h
55.32	425D47AEh	42h	5Dh	47h	AEh
2.0	40000000h	40h	00h	00h	00h
1.0	3F800000h	3Fh	80h	00h	00h
-1.0	BF800000h	BFh	80h	00h	00h

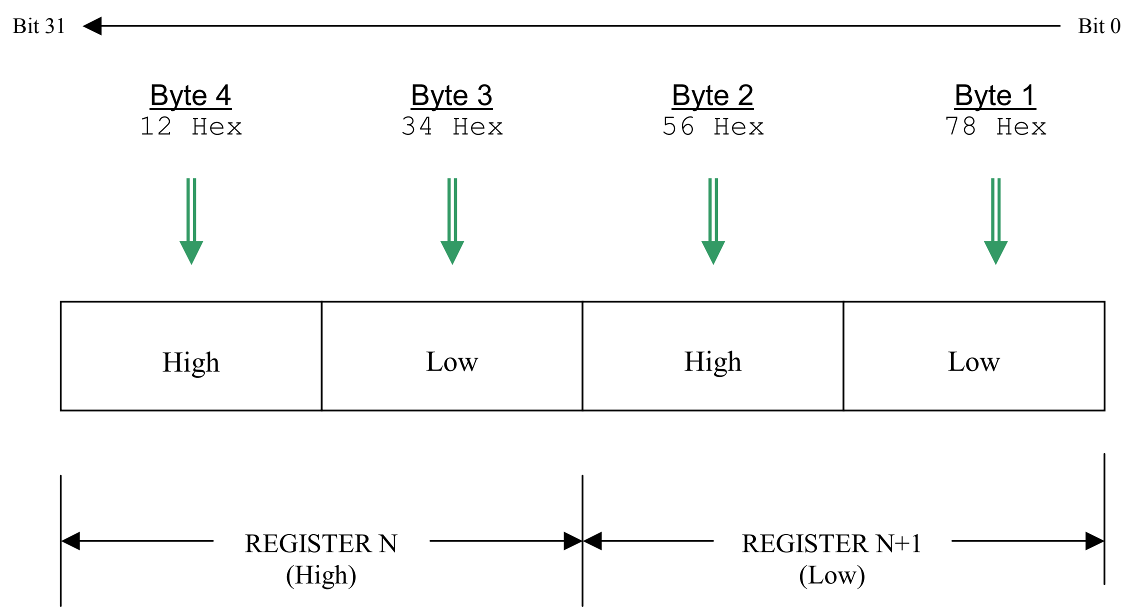
3.2 Unsigned/signed 32-bit Register Formats

The formats descriptions below use the value 12345678 Hex as an example. Where the binary representation is:

	Byte 4 = 12 Hex								**Byte 3 = 34 Hex**								**Byte 2 = 56 Hex**								**Byte 1 = 78 Hex**							
Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0

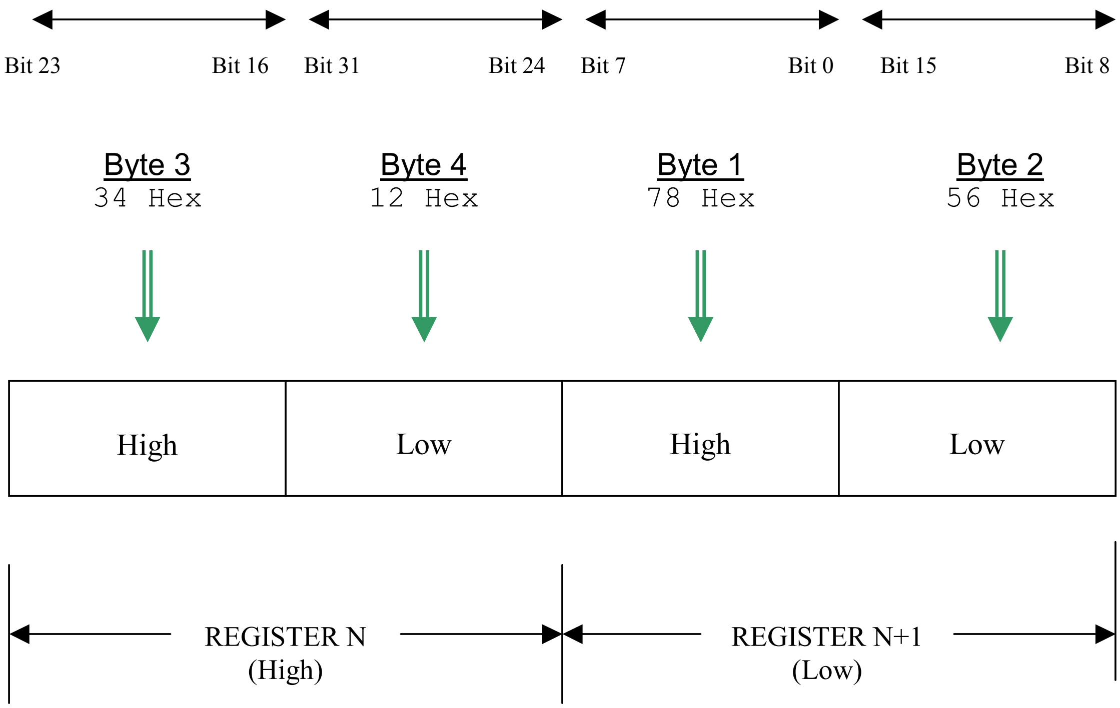
FP B – Big Endian Format

The value 12345678 Hex will be represented as follows:



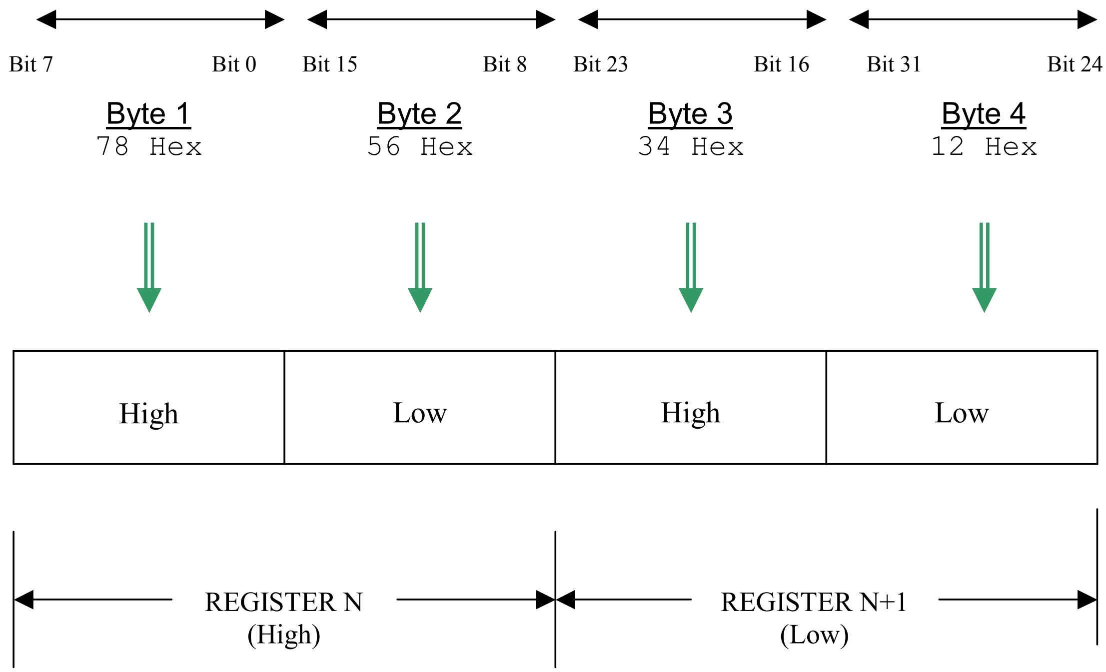
FP BB – Big Endian Byte Swapped Format

The value 12345678 Hex will be represented as follows:



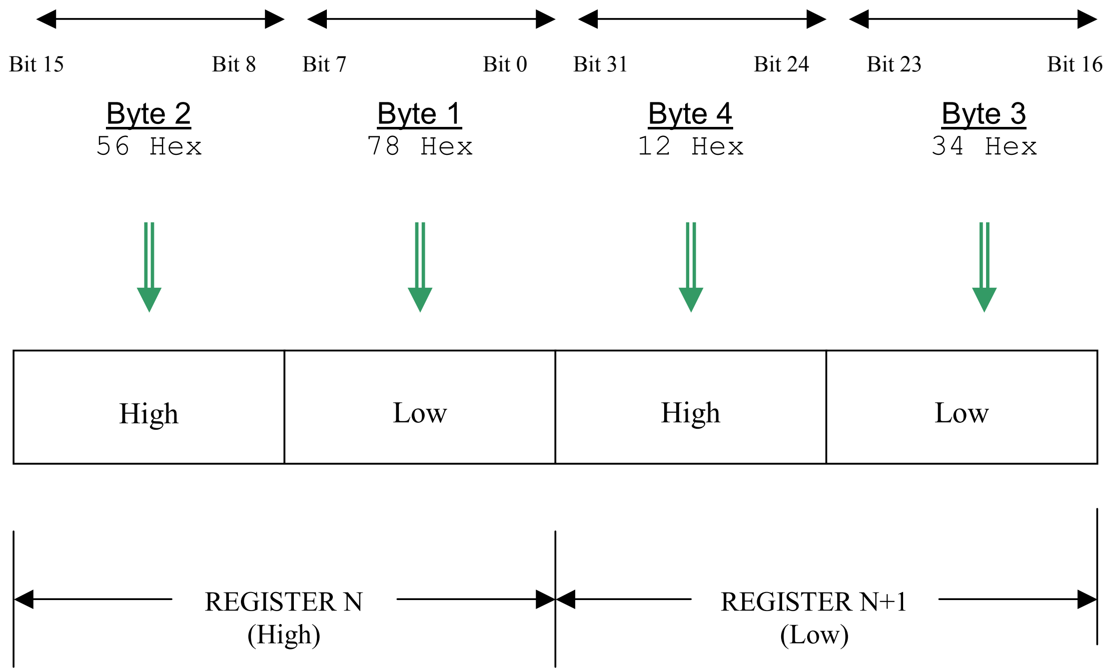
FP L – Little Endian Format

The value 12345678 Hex will be represented as follows:



FP LB – Little Endian Byte Swap Format

The value 12345678 Hex will be represented as follows:



4. Modbus/TCP & Modbus RTU Function Codes

4.1 Definition

The HC900 Modbus protocol uses a subset of the standard Modbus function codes to provide access to process-related information. These standard function codes provide basic support for IEEE 32-bit floating point numbers, 32-bit unsigned/signed integer and 16-bit integer register representation of instrument's process data.

Table 4-1, Table 4-2, and Table 4-3 list the Function Code definitions, the maximum number of Object Addresses and maximum number of registers allowed per request. Also shown are differences between firmware versions 2.3 and 2.4.

Repolling of data is not supported by this instrument.

Table 4-1 Modbus/TCP and Modbus RTU Function Codes Definitions

Function Code	Name	Usage
01	Read Coil Status	Read the state of a digital output
02	Read Input Status	Read the state of a digital input
03	Read Holding Registers	Read data in 16-bit Register Format (high/low). Used to read integer or floating point process data. Registers are consecutive and are imaged from the instrument to the host.
04	Read Input Registers	Provides Read access to any Analog Input Channel positioned in any Rack or Slot.
05	Force Single Coil	Write data to force a digital output ON/OFF Values of FF 00 forces digital output ON Values of 00 00 forces digital output OFF Values of FF FF releases the force of the digital output All other values are illegal and will not effect the digital output.
06	Preset Single Register	Write Data in 16-bit Integer Format (high/low) ONLY.
08	Loopback Test	Used for diagnostic testing of the communications port.
16 (10h)	Preset Multiple Registers	Write Data in 16-bit Format (high/low). Used to write integer and floating point override data. Registers are consecutive and are imaged from the host to the instrument.
17 (11h)	Report Device ID	Read instrument ID and connection information, ROM version, etc.

Table 4-2 Maximum Number of Object Addresses

Object Name	Max. No. of Addresses		Function Code
	C30	C50/C70/C70R	
Analog Inputs	96/12 slots v2.3 192/12 slots v2.4	640 v2.3 1280 v2.4	3: can only access first 8 slots of rack 1 4: can access all slots and racks
Discrete Input	192/12 slots v2.3 384/12 slots v2.4	1280 v2.3 2560 v2.4	2
Discrete Output/Coil	192/12 slots v2.3 384/12 slots v2.4	1280 v2.3 2560 v2.4	1: read 5: force
Loop	8	32	3
Variable Value	600	600	3
Set Point Programmer Value	8	8	3
Segments per Set Point Programmer	50	50	3
Tagged Signals	2000	2000 (C50) 5000 (C70/C70R)	3
Scheduler Value	2	2	3
Segments per Scheduler Schedule	50	50	3
Sequencer	4	4	3
Stage Group	8	8	3
Ramp Group	8	8	3
Hand-Off-Auto	16	16	3
Alternator	6	6	3
Device Control	16	16	3
User Defined Registers	1024	1024	3

Table 4-3 Maximum Number of Registers Allowable per Request

Function Code	Max. No. of Registers
1, 2	2040 bits
3, 4	127 Registers 63 Floats
5	1 Coil
6	1 Register
10h	127 Registers 63 Floats

4.2 Function Code 01 – Read Digital Output Status

Description

Function code 01 (0X references) is used to read a digital output's ON/OFF status of the HC900 using 16 bit addressing for DO access and data is returned in a binary format mapped into bytes.

The Modbus Comm Digital I/O Channel to address mapping is shown starting on page 20.

Broadcast is not supported.

Query

The query message specifies the starting Digital Output (DO) and the quantity of DOs to read. The DO address in the message is based on the rack slot and channel number of the digital output being read.

Example Query: Read DO channels 1 to 16, located in Rack #1, Slot #1; from the controller with slave address 1.

Query message format for function code 01

	Slave Address (00 for TCP)	Function Code	Starting Address High	Starting Address Low	Number DO High	Number DO Low	CRC (RTU)	CRC (RTU)
TCP Example	00	01	00	00	00	10		
RTU Example	01	01	00	00	00	10	CRC	CRC

Response

The DO status in the response message is packed as one DO per bit of the data field. Status is indicated as: 1 = ON; 0 = OFF. The LSB of the first data byte contains the DO addressed in the query. The other DOs follow toward the high order end of this byte, and from low order to high order in subsequent bytes.

If the returned DO quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros (toward the high order end of the byte). The byte count field specifies the quantity of data bytes returned.

Example Response: DO channels 2 and 6 located in Rack #1, Slot #1 are on; all others are off.

Response message format for function code 01

	Slave Address (00 for TCP)	Function Code	Byte Count	Data	Data	CRC (RTU)	CRC (RTU)
TCP Example	00	01	02	22	00		
RTU Example	01	01	02	22	00	CRC	CRC

In the response the status of DOs 1 - 8 is shown as the byte value 22 hex, or 0010 0010 binary. DO 8 is the MSB of this byte, and DO 1 is the LSB. Left to right, the status of DO 8 through 1 is: OFF-OFF-ON-OFF-OFF-OFF-ON-OFF. The status of DOs 9 - 16 are shown a 00hex, or 0000 0000 with the same bit ordering.

Digital I/O Channel to Address Mapping

For HC900 CPUs and Scanners with firmware version 2.3 and earlier each rack is allocated addressing for a maximum of 16 slots with 16 channels. (Note: Up to 16 slots are accommodated in the protocol even though the largest rack available supports 12 slots.) For version 2.4 each rack is allocated addressing for a maximum of 16 slots with 32 channels. Each slot can be a DI or DO card.

Table 4-4 through Table 4-7 define the rack, slot and channel address mapping used for DI/DO. Each DI/DO consumes 1 Modbus bit address.

Decimal addressing is typically non-zero based for DI/DO access (1-based), applicable to coil or register address.

Table 4-4 DI/DO Address Mapping (version 2.3 and earlier)

Rack	Channels	Coil number/ Register number	Modbus Hex Address Range
1	1 - 256	1 – 256	0 - FF
2	257 - 512	257 - 512	100 - 1FF
3	513 - 768	513 – 768	200 - 2FF
4	769 - 1024	769 – 1024	300 - 3FF
5	1025 - 1280	1025 - 1280	400 - 4FF

Note: if a high-density card (32 channels) is inserted in the rack, only the first 16 channels of the card can be accessed. The high-density map in Table 4-6 must be used to access all of the card's channels.

The coil (register) number for a digital I/O is based on the DI/DO's position in the card cage. It is determined from the formula:

Coil (register) Number = [(Rack-1)*256] + [(Slot-1)*16] + channel in module

Example: To monitor a coil (register) located in the 2nd channel of slot 10 of rack 3, the Modbus coil (register) number is:

[(3-1)*256] + [(10-1)*16] + 2 = 658

Some third party software packages will require the 1-based coil/register number to be used for the address while others will require the 0-based hex address.

Table 4-5 shows the Modbus Comm Digital I/O Channel to Address Mapping for Rack #1, firmware version 2.3 and earlier. Refer to Table 4-4 for Address Ranges for Racks #2 through #5. Note: if a high-density card (32 channels) is inserted in the rack, only the first 16 channels of the card can be accessed. The high-density map in Table 4-6 must be used to access all of the card's channels.

Table 4-5 Modbus Comm Digital I/O Channel to Address Mapping - Rack #1 (v2.3 and earlier)

Slot 1			Slot 2			Slot 3			Slot 4		
CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex
16	16	0F	16	32	1F	16	48	2F	16	64	3F
15	15	0E	15	31	1E	15	47	2E	15	63	3E
14	14	0D	14	30	1D	14	46	2D	14	62	3D
13	13	0C	13	29	1C	13	45	2C	13	61	3C
12	12	0B	12	28	1B	12	44	2B	12	60	3B
11	11	0A	11	27	1A	11	43	2A	11	59	3A
10	10	9	10	26	19	10	42	29	10	58	39
9	9	8	9	25	18	9	41	28	9	57	38
8	8	7	8	24	17	8	40	27	8	56	37
7	7	6	7	23	16	7	39	26	7	55	36
6	6	5	6	22	15	6	38	25	6	54	35
5	5	4	5	21	14	5	37	24	5	53	34
4	4	3	4	20	13	4	36	23	4	52	33
3	3	2	3	19	12	3	35	22	3	51	32
2	2	1	2	18	11	2	34	21	2	50	31
1	1	0	1	17	10	1	33	20	1	49	30

Slot 5			Slot 6			Slot 7			Slot 8		
CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex
16	80	4F	16	96	5F	16	112	6F	16	128	7F
15	79	4E	15	95	5E	15	111	6E	15	127	7E
14	78	4D	14	94	5D	14	110	6D	14	126	7D
13	77	4C	13	93	5C	13	109	6C	13	125	7C
12	76	4B	12	92	5B	12	108	6B	12	124	7B
11	75	4A	11	91	5A	11	107	6A	11	123	7A
10	74	49	10	90	59	10	106	69	10	122	79
9	73	48	9	89	58	9	105	68	9	121	78
8	72	47	8	88	57	8	104	67	8	120	77
7	71	46	7	87	56	7	103	66	7	119	76
6	70	45	6	86	55	6	102	65	6	118	75
5	69	44	5	85	54	5	101	64	5	117	74
4	68	43	4	84	53	4	100	63	4	116	73
3	67	42	3	83	52	3	99	62	3	115	72
2	66	41	2	82	51	2	98	61	2	114	71
1	65	40	1	81	50	1	97	60	1	113	70

Slot 9		
CH#	Coil/ register	Addr. Hex
16	144	8F
15	143	8E
14	142	8D
13	141	8C
12	140	8B
11	139	8A
10	138	89
9	137	88
8	136	87
7	135	86
6	134	85
5	133	84
4	132	83
3	131	82
2	130	81
1	129	80

Slot 10		
CH#	Coil/ register	Addr. Hex
16	160	9F
15	159	9E
14	158	9D
13	157	9C
12	156	9B
11	155	9A
10	154	99
9	153	98
8	152	97
7	151	96
6	150	95
5	149	94
4	148	93
3	147	92
2	146	91
1	145	90

Slot 11		
CH#	Coil/ register	Addr. Hex
16	176	AF
15	175	AE
14	174	AD
13	173	AC
12	172	AB
11	171	AA
10	170	A9
9	169	A8
8	168	A7
7	167	A6
6	166	A5
5	165	A4
4	164	A3
3	163	A2
2	162	A1
1	161	A0

Slot 12		
CH#	Coil/ register	Addr. Hex
16	192	BF
15	191	BE
14	190	BD
13	189	BC
12	188	BB
11	187	BA
10	186	B9
9	185	B8
8	184	B7
7	183	B6
6	182	B5
5	181	B4
4	180	B3
3	179	B2
2	178	B1
1	177	B0

Slot 13		
CH#	Coil/ register	Addr. Hex
16	208	CF
15	207	CE
14	206	CD
13	205	CC
12	204	CB
11	203	CA
10	202	C9
9	201	C8
8	200	C7
7	199	C6
6	198	C5
5	197	C4
4	196	C3
3	195	C2
2	194	C1
1	193	C0

Slot 14		
CH#	Coil/ register	Addr. Hex
16	224	DF
15	223	DE
14	222	DD
13	221	DC
12	220	DB
11	219	DA
10	218	D9
9	217	D8
8	216	D7
7	215	D6
6	214	D5
5	213	D4
4	212	D3
3	211	D2
2	210	D1
1	209	D0

Slot 15		
CH#	Coil/ register	Addr. Hex
16	240	EF
15	239	EE
14	238	ED
13	237	EC
12	236	EB
11	235	EA
10	234	E9
9	233	E8
8	232	E7
7	231	E6
6	230	E5
5	229	E4
4	228	E3
3	227	E2
2	226	E1
1	225	E0

Slot 16		
CH#	Coil/ register	Addr. Hex
16	256	FF
15	255	FE
14	254	FD
13	253	FC
12	252	FB
11	251	FA
10	250	F9
9	249	F8
8	248	F7
7	247	F6
6	246	F5
5	245	F4
4	244	F3
3	243	F2
2	242	F1
1	241	F0

Table 4-6 DI/DO Address Mapping (v2.4)

Rack	Channels	Coil number/register number	Modbus Hex Address Range
1	1 - 512	2001 – 2512	7D0 – 9CF
2	513 – 1024	2513 – 3024	9D0 – BCF
3	1025 – 1536	3025 – 3536	BD0 – DCF
4	1537 - 2048	3537 – 4048	DD0 – FCF
5	2049 - 2560	4049 - 4560	FD0 – 11CF

Table 4-7 shows the Modbus Comm Digital I/O Channel to Address Mapping for Rack #1 version 2.4. Refer to Table 4-6 for Address Ranges for Racks #2 through #5.

The coil (register) number for a DI/DO is based on the DI/DO's position in the card cage. It is determined from the formula:

Coil (register) Number = [(Rack-1)*512] + [(Slot-1)*32] + channel in module + 2000

Example: To monitor a coil (register) located in the 2nd channel of slot 10 of rack 3, the Modbus coil (register) number is:

[(3-1)*512] + [(10-1)*32] + 2 + 2000 = 3314

Some third party software packages will require the 1-based coil/register number to be used for the address while others will require the 0-based hex address.

Table 4-7 Modbus Comm Digital I/O Channel to Address Mapping - Rack #1 (version 2.4)

Slot 1			Slot 2			Slot 3			Slot 4		
CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex
32	2032	7EF	32	2064	80F	32	2096	82F	32	2128	84F
31	2031	7EE	31	2063	80E	31	2095	82E	31	2127	84E
30	2030	7ED	30	2062	80D	30	2094	82D	30	2126	84D
29	2029	7EC	29	2061	80C	29	2093	82C	29	2125	84C
28	2028	7EB	28	2060	80B	28	2092	82B	28	2124	84B
27	2027	7EA	27	2059	80A	27	2091	82A	27	2123	84A
26	2026	7E9	26	2058	809	26	2090	829	26	2122	849
25	2025	7E8	25	2057	808	25	2089	828	25	2121	848
24	2024	7E7	24	2056	807	24	2088	827	24	2120	847
23	2023	7E6	23	2055	806	23	2087	826	23	2119	846
22	2022	7E5	22	2054	805	22	2086	825	22	2118	845
21	2021	7E4	21	2053	804	21	2085	824	21	2117	844
20	2020	7E3	20	2052	803	20	2084	823	20	2116	843
19	2019	7E2	19	2051	802	19	2083	822	19	2115	842
18	2018	7E1	18	2050	801	18	2082	821	18	2114	841
17	2017	7E0	17	2049	800	17	2081	820	17	2113	840
16	2016	7DF	16	2048	7FF	16	2080	81F	16	2112	83F
15	2015	7DE	15	2047	7FE	15	2079	81E	15	2111	83E
14	2014	7DD	14	2046	7FD	14	2078	81D	14	2110	83D
13	2013	7DC	13	2045	7FC	13	2077	81C	13	2109	83C
12	2012	7DB	12	2044	7FB	12	2076	81B	12	2108	83B
11	2011	7DA	11	2043	7FA	11	2075	81A	11	2107	83A
10	2010	7D9	10	2042	7F9	10	2074	819	10	2106	839
9	2009	7D8	9	2041	7F8	9	2073	818	9	2105	838
8	2008	7D7	8	2040	7F7	8	2072	817	8	2104	837
7	2007	7D6	7	2039	7F6	7	2071	816	7	2103	836
6	2006	7D5	6	2038	7F5	6	2070	815	6	2102	835
5	2005	7D4	5	2037	7F4	5	2069	814	5	2101	834
4	2004	7D3	4	2036	7F3	4	2068	813	4	2100	833
3	2003	7D2	3	2035	7F2	3	2067	812	3	2099	832
2	2002	7D1	2	2034	7F1	2	2066	811	2	2098	831
1	2001	7D0	1	2033	7F0	1	2065	810	1	2097	830

Slot 5		
CH#	Coil/ register	Addr. Hex
32	2160	86F
31	2159	86E
30	2158	86D
29	2157	86C
28	2156	86B
27	2155	86A
26	2154	869
25	2153	868
24	2152	867
23	2151	866
22	2150	865
21	2149	864
20	2148	863
19	2147	862
18	2146	861
17	2145	860
16	2144	85F
15	2143	85E
14	2142	85D
13	2141	85C
12	2140	85B
11	2139	85A
10	2138	859
9	2137	858
8	2136	857
7	2135	856
6	2134	855
5	2133	854
4	2132	853
3	2131	852
2	2130	851
1	2129	850

Slot 6		
CH#	Coil/ register	Addr. Hex
32	2192	88F
31	2191	88E
30	2190	88D
29	2189	88C
28	2188	88B
27	2187	88A
26	2186	889
25	2185	888
24	2184	887
23	2183	886
22	2182	885
21	2181	884
20	2180	883
19	2179	882
18	2178	881
17	2177	880
16	2176	87F
15	2175	87E
14	2174	87D
13	2173	87C
12	2172	87B
11	2171	87A
10	2170	879
9	2169	878
8	2168	877
7	2167	876
6	2166	875
5	2165	874
4	2164	873
3	2163	872
2	2162	871
1	2161	870

Slot 7		
CH#	Coil/ register	Addr. Hex
32	2224	8AF
31	2223	8AE
30	2222	8AD
29	2221	8AC
28	2220	8AB
27	2219	8AA
26	2218	8A9
25	2217	8A8
24	2216	8A7
23	2215	8A6
22	2214	8A5
21	2213	8A4
20	2212	8A3
19	2211	8A2
18	2210	8A1
17	2209	8A0
16	2208	89F
15	2207	89E
14	2206	89D
13	2205	89C
12	2204	89B
11	2203	89A
10	2202	899
9	2201	898
8	2200	897
7	2199	896
6	2198	895
5	2197	894
4	2196	893
3	2195	892
2	2194	891
1	2193	890

Slot 8		
CH#	Coil/ register	Addr. Hex
32	2256	8D0
31	2255	8CE
30	2254	8CD
29	2253	8CC
28	2252	8CB
27	2251	8CA
26	2250	8C9
25	2249	8C8
24	2248	8C7
23	2247	8C6
22	2246	8C5
21	2245	8C4
20	2244	8C3
19	2243	8C2
18	2242	8C1
17	2241	8C0
16	2240	8BF
15	2239	8BE
14	2238	8BD
13	2237	8BC
12	2236	8BB
11	2235	8BA
10	2234	8B9
9	2233	8B8
8	2232	8B7
7	2231	8B6
6	2230	8B5
5	2229	8B4
4	2228	8B3
3	2227	8B2
2	2226	8B1
1	2225	8B0

Slot 9		
CH#	Coil/ register	Addr. Hex
32	2288	8EF
31	2287	8EE
30	2286	8ED
29	2285	8EC
28	2284	8EB
27	2283	8EA
26	2282	8E9
25	2281	8E8
24	2280	8E7
23	2279	8E6
22	2278	8E5
21	2277	8E4
20	2276	8E3
19	2275	8E2
18	2274	8E1
17	2273	8E0
16	2272	8DF
15	2271	8DE
14	2270	8DD
13	2269	8DC
12	2268	8DB
11	2267	8DA
10	2266	8D9
9	2265	8D8
8	2264	8D7
7	2263	8D6
6	2262	8D5
5	2261	8D4
4	2260	8D3
3	2259	8D2
2	2258	8D1
1	2257	8D0

Slot 10		
CH#	Coil/ register	Addr. Hex
32	2320	90F
31	2319	90E
30	2318	90D
29	2317	90C
28	2316	90B
27	2315	90A
26	2314	909
25	2313	908
24	2312	907
23	2311	906
22	2310	905
21	2309	904
20	2308	903
19	2307	902
18	2306	901
17	2305	900
16	2304	8FF
15	2303	8FE
14	2302	8FD
13	2301	8FC
12	2300	8FB
11	2299	8FA
10	2298	8F9
9	2297	8F8
8	2296	8F7
7	2295	8F6
6	2294	8F5
5	2293	8F4
4	2292	8F3
3	2291	8F2
2	2290	8F1
1	2289	8F0

Slot 11		
CH#	Coil/ register	Addr. Hex
32	2352	92F
31	2351	92E
30	2350	92D
29	2349	92C
28	2348	92B
27	2347	92A
26	2346	929
25	2345	928
24	2344	927
23	2343	926
22	2342	925
21	2341	924
20	2340	923
19	2339	922
18	2338	921
17	2337	920
16	2336	91F
15	2335	91E
14	2334	91D
13	2333	91C
12	2332	91B
11	2331	91A
10	2330	919
9	2329	918
8	2328	917
7	2327	916
6	2326	915
5	2325	914
4	2324	913
3	2323	912
2	2322	911
1	2321	910

Slot 12		
CH#	Coil/ register	Addr. Hex
32	2384	94F
31	2383	94E
30	2382	94D
29	2381	94C
28	2380	94B
27	2379	94A
26	2378	949
25	2377	948
24	2376	947
23	2375	946
22	2374	945
21	2373	944
20	2372	943
19	2371	942
18	2370	941
17	2369	940
16	2368	93F
15	2367	93E
14	2366	93D
13	2365	93C
12	2364	93B
11	2363	93A
10	2362	939
9	2361	938
8	2360	937
7	2359	936
6	2358	935
5	2357	934
4	2356	933
3	2355	932
2	2354	931
1	2353	930

Slot 13			Slot 14			Slot 15			Slot 16		
CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex	CH#	Coil/ register	Addr. Hex
32	2416	96F	32	2448	98F	32	2480	9AF	32	2512	9CF
31	2415	96E	31	2447	98E	31	2479	9AE	31	2511	9CE
30	2414	96D	30	2446	98D	30	2478	9AD	30	2510	9CD
29	2413	96C	29	2445	98C	29	2477	9AC	29	2509	9CC
28	2412	96B	28	2444	98B	28	2476	9AB	28	2508	9CB
27	2411	96A	27	2443	98A	27	2475	9AA	27	2507	9CA
26	2410	969	26	2442	989	26	2474	9A9	26	2506	9C9
25	2409	968	25	2441	988	25	2473	9A8	25	2505	9C8
24	2408	967	24	2440	987	24	2472	9A7	24	2504	9C7
23	2407	966	23	2439	986	23	2471	9A6	23	2503	9C6
22	2406	965	22	2438	985	22	2470	9A5	22	2502	9C5
21	2405	964	21	2437	984	21	2469	9A4	21	2501	9C4
20	2404	963	20	2436	983	20	2468	9A3	20	2500	9C3
19	2403	962	19	2435	982	19	2467	9A2	19	2499	9C2
18	2402	961	18	2434	981	18	2466	9A1	18	2498	9C1
17	2401	960	17	2433	980	17	2465	9A0	17	2497	9C0
16	2400	95F	16	2432	97F	16	2464	99F	16	2496	9BF
15	2399	95E	15	2431	97E	15	2463	99E	15	2495	9BE
14	2398	95D	14	2430	97D	14	2462	99D	14	2494	9BD
13	2397	95C	13	2429	97C	13	2461	99C	13	2493	9BC
12	2396	95B	12	2428	97B	12	2460	99B	12	2492	9BB
11	2395	95A	11	2427	97A	11	2459	99A	11	2491	9BA
10	2394	959	10	2426	979	10	2458	999	10	2490	9B9
9	2393	958	9	2425	978	9	2457	998	9	2489	9B8
8	2392	957	8	2424	977	8	2456	997	8	2488	9B7
7	2391	956	7	2423	976	7	2455	996	7	2487	9B6
6	2390	955	6	2422	975	6	2454	995	6	2486	9B5
5	2389	954	5	2421	974	5	2453	994	5	2485	9B4
4	2388	953	4	2420	973	4	2452	993	4	2484	9B3
3	2387	952	3	2419	972	3	2451	992	3	2483	9B2
2	2386	951	2	2418	971	2	2450	991	2	2482	9B1
1	2385	950	1	2417	970	1	2449	990	1	2481	9B0

4.3 Function Code 02 - Read Digital Input Status

Description

Function code 02 (1X references) is used to read a digital input's ON/OFF status using 16 bit addressing for DI access and data is returned in a binary format mapped into bytes.

The Modbus Comm Digital I/O Channel to address mapping is shown starting on page 20.

Broadcast is not supported.

Query

The query message specifies the starting input and the quantity of inputs to read. The DI address in the message is based on the slot and channel number of the digital input being read.

Example: Read inputs for channels 1 to 16 in Rack #1, Slot 1, from the controller with slave address 1.

Query message format for function code 02

	Slave Address (00 for TCP)	Function Code	Starting Address High	Starting Address Low	Number Inputs High	Number Inputs Low	CRC (RTU)	CRC (RTU)
TCP Example	00	02	00	00	00	10		
RTU Example	01	02	00	00	00	10	CRC	CRC

Response

The input status in the response message is packed as one input per bit of the data field. Status is indicated as: 1 = ON; 0 = OFF. The LSB of the first data byte contains the input addressed in the query. The other inputs follow toward the high order end of this byte, and from low order to high order in subsequent bytes.

If the returned input quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros (toward the high order end of the byte). The byte count field specifies the quantity of data bytes returned.

Example: Inputs for channels 2 and 6 in Rack #1, Slot 1 are on, all others are off.

Response message format for function code 02

	Slave Address (00 for TCP)	Function Code	Byte Count	Data	Data	CRC (RTU)	CRC (RTU)
TCP Example	00	02	02	22	00		
RTU Example	01	02	02	22	00	CRC	CRC

In the response the status of inputs 1 - 8 is shown as the byte value 22 hex, or 0010 0010 binary. Input 8 is the MSB of this byte, and input 1 is the LSB. Left to right, the status of input 6 through 1 is: OFF-OFF-ON-OFF-OFF-OFF-ON-OFF. The status of inputs 9-16 are shown as 00 hex, or 0000 0000 with the same bit ordering.

4.4 Function Codes 03- Read Holding (Data) Registers

Description

Function code 03 (also referred to as 4X decimal references) is used to read 32-bit floating point, 32-bit unsigned/signed integer and 16 bit integer data in the controller as described in Section 6. Registers are consecutive.

It is also used to Read certain analog input modules for commonality of UMC800 controller addresses. This applies to analog input modules positioned ONLY in the first 8 Slots of Rack #1, providing support ONLY for the first 64 channels. **Use Function Code 04 to address all analog inputs in the HC900 controller.**

Table 4-8 HC900 AI Address Mapping supported by Function Code 03

Rack	Channel	Register Range	Hex Address Range
1	1 - 64	1 - 127	0 -7F (uses register addressing 1800 - 187Fh)
	65 - 128	65 - 255	Not Supported

If a request is made to an address that does not exist in the map in Section 6 , the controller will honor that request and return zeros for that address. This behavior will greatly enhance the bandwidth on the link vs. making several different requests for non-contiguous data elements. (i.e. Consider a controller that is configured for AI #1 and AI #3 and for some reason AI #2 is an invalid request.) The contiguous method would allow the read of AI #1 through AI #3 and the data location for AI #2 would be zeros.

Broadcast is not supported.

Query

The query message specifies the starting register and quantity of registers to be read. Registers are addressed starting at zero: registers 1-16 are addressed as 0-15.

Example: Read PV, Remote SP, Working SP, and Output as floating point values for Loop #1 in the controller at slave address 1.

Query message format for function code 03

	Slave Address (00 for TCP)	Function Code	Starting Address High	Starting Address Low	Number Addresses High	Number Addresses Low	CRC (RTU)	CRC (RTU)
TCP Example	00	03	00	40	00	08		
RTU Example	01	03	00	40	00	08	CRC	CRC

Response

The register data in the response message is packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits.

The floating point values require two consecutive registers. The byte order of the floating point number is determined by the setting of the byte swap configuration value. In this example, and the examples that follow, the byte swap order is FP B. Refer to page 10. The first 16 bits of the response contain the IEEE MSB of the float value. The second 16 bits of the response contain the IEEE LSB of the float value. If the master station requests only one register at an address of a floating point value then half of a float will be returned.

The Modbus RTU protocol has a single byte count for function code 03, therefore the Modbus RTU protocol can only process up to 63 floating point and 127 16-bit integer values in a single request.

Example: PV, Remote SP, Working SP, and Output where PV=100.0, RSP=100.0, WSP=100.0, and Out=55.32 as floating point values where AI #1 = 100.0 and AI #2 = 55.32

Response message format for function codes 03

	Slave Address (00 for TCP)	Function Code	Byte Count	Data	Data	Data	Data	CRC (RTU)	CRC (RTU)
TCP Example	00	03	10	42 C8 00 00 (100)	42 C8 00 00 (100)	42 C8 00 00 (100)	42 5D 47 AE (55.32)		
RTU Example	01	03	10	42 C8 00 00 (100)	42 C8 00 00 (100)	42 C8 00 00 (100)	42 5D 47 AE (55.32)	CRC	CRC

4.5 Function Code 04 - Read Input Registers

Description

Function code 04(3X references) provides read access to Analog Input modules positioned in ANY Rack or Slot. All values are in IEEE 32-bit floating point format.

For HC900 CPUs and Scanners with firmware version 2.3 and earlier, each Rack is allocated space for a maximum of 16 Slots and each Slot assumes Modules with a maximum of 8 Channels, which consumes 16 Modbus Register addresses. See Table 4-9.

For HC900 CPUs and Scanners with firmware version 2.4, each Rack is allocated space for a maximum of 16 Slots and each Slot assumes Modules with a maximum of 16 Channels, which consumes 32 Modbus Register addresses. See Table 4-10.

In Table 4-9 and Table 4-10 the register addressing is 1-based while the hex addressing is 0-based.

Table 4-9 HC900 AI Address Mapping supported by Function Code 04 (v2.3)

Rack	Channels	Register Range (decimal)	Hex Address Range
1	1 - 128	1 - 255	0 –FE
2	129 - 256	257 - 511	100 - 1FE
3	257 - 384	513 - 767	200 - 2FE
4	385 - 512	769 - 1023	300 - 3FE
5	513 - 640	1025 - 1279	400 - 4FE

Table 4-10 HC900 AI Address Mapping supported by Function Code 04 (v2.4)

Rack	Channels	Register Range (decimal)	Hex Address Range
1	1 – 256	2001 – 2511	0 -9CE
2	257 - 512	2513 – 3023	9D0 – BCE
3	513 - 768	3025 – 3535	BD0 – DCE
4	769 - 1024	3537 – 4047	DD0 – FCE
5	1025 - 1280	4049 - 4559	FD0 – 11CE

If a request is made to an address that does not exist in the map in Section 1, the controller will honor that request and return zeros for that address. This behavior will greatly enhance the bandwidth on the link vs. making several different requests for non-contiguous data elements. (i.e. Consider a controller that is configured for AI #1 and AI #3 and for some reason AI #2 is an invalid request.) The contiguous method would allow the read of AI #1 through AI #3 and the data location for AI #2 would be zeros.

Broadcast is not supported.

Query

The query message specifies the starting register and quantity of registers to be read. Registers are addressed starting at zero: registers 1-16 are addressed as 0-15.

Example: Read analog inputs #1 and #2 (Rack #1, Module #1) addresses 0-3, as floating point values from the controller at slave address 1.

Query message format for function code 04

	Slave Address (00 for TCP)	Function Code	Starting Address High	Starting Address Low	Number Addresses High	Number Addresses Low	CRC (RTU)	CRC (RTU)
TCP Example	00	04	00	00	00	04		
RTU Example	01	04	00	00	00	04	CRC	CRC

Response

The register data in the response message is packed as two bytes per register. For each register, the first byte contains the high order bits and the second contains the low order bits.

The floating point values require two consecutive registers. The byte order of the floating point number is determined by the setting of the byte swap configuration value. In this example, and the examples that follow, the byte swap order is FP B. Refer to subsection 1.3. The first 16 bits of the response contain the IEEE MSB of the float value. The second 16 bits of the response contain the IEEE LSB of the float value. If the master station requests only one register at an address of a floating point value, then half of a float will be returned.

The Modbus RTU protocol has a single byte count for function code 04, therefore the Modbus RTU protocol can only process up to 63 floating point values in a single request.

Example: Analog inputs #1 and #2 as floating point values where AI #1 = 100.0 and AI #2 =55.32

Response message format for function codes 04

	Slave Address (00 for TCP)	Function Code	Byte Count	Data	Data	CRC (RTU)	CRC (RTU)
TCP Example	00	04	08	42 C8 00 00 (100)	42 5D 47 AE (55.32)		
RTU Example	01	04	08	42 C8 00 00 (100)	42 5D 47 AE (55.32)	CRC	CRC

4.6 Function Code 05 - Force Single Digital Output

Description

Force a single digital output (0X reference) to either ON or OFF. These are the same digital outputs (DO) used in Function Code 01.
The Modbus Comm Digital I/O Channel to address mapping is shown starting on page 20.

The HC900 does not support broadcast, and forcing can only be done in the Run mode.

Query

The query message specifies the DO to be forced. Registers are addressed starting at zero:
DO 1 is bit address 0.

The requested ON/OFF state is specified by a constant in the query data field.
A value of FF 00 hex requests it to be ON.
A value of 00 00 hex requests it to be OFF.

A value of FF FF releases the force.

ATTENTION: Any query (ON or OFF) causes a force mode of this point in the HC900 controller. The Green force LED goes ON. While in this mode, internal control of function blocks cannot communicate to this point. **DON'T FORGET** to send a query to release this force.

Example: Force DO Card Rack #1, Slot 1, Channel 6 ON in the controller at slave address 1.

Query message format for function code 05

	Slave Address (00 for TCP)	Function Code	DO Address High	DO Address Low	Force Data High	Force Data Low	CRC (RTU)	CRC (RTU)
TCP Example	00	05	00	05	FF	00		
RTU Example	01	05	00	05	FF	00	CRC	CRC

Response

The normal response is an echo of the query, returned after the DO state has been forced.

Example: Force DO Card Rack #1, Slot 1, Channel 6 ON in the controller at slave address 1.

Response message format for function code 05

	Slave Address (00 for TCP)	Function Code	DO Address High	DO Address Low	Force Data High	Force Data Low	CRC (RTU)	CRC (RTU)
TCP Example	00	05	00	05	FF	00		
RTU Example	01	05	00	05	FF	00	CRC	CRC

The Modbus Comm Digital I/O Channel to address mapping is shown starting on page 20.

4.7 Function Code 06 - Preset Single Register

Description

Presets integer value into a single register (also refered to as 4X references).

The HC900 does not support Broadcast.

The registers that are specified in Section 6 with an access type "W" and integer and bit packed (16-bit register) data types, can be written to via Function Code 06. Also, digital variables in 32-bit floating point format can be written using this function code. Writing a non-zero value to either register of the digital variable will result in a floating point 1 to be written to the variable. Writing a zero value to either register will result in the variable being set to 0.

Query

The query message specifies the register references to be preset. Registers are addressed starting at zero: Register 1 is addressed as 0.

Example: Set Rack #1, Loop #1 to Auto (address 00FAh) in the controller at slave address 1

Query message format for function code 06

	Slave Address (00 for TCP)	Function Code	Address High	Address Low	Preset Data High	Preset Data Low	CRC (RTU)	CRC (RTU)
TCP Example	00	06	00	FA	00	01		
RTU Example	01	06	00	FA	00	01	CRC	CRC

Response

The normal response is an echo of the query returned after the register contents have been preset.

Example: Set Rack #1, Loop #1 to Auto (address 00FAh) in the controller at slave address 1.

Response message format for function code 06

	Slave Address (00 for TCP)	Function Code	Address High	Address Low	Preset Data High	Preset Data Low	CRC (RTU)	CRC (RTU)
TCP Example	00	06	00	FA	00	01		
RTU Example	01	06	00	FA	00	01	CRC	CRC

4.8 Function Code 08 - Loopback Message

Description

Echoes received query message.

Query

Message can be any length up to half the length of the data buffer minus 8 bytes.

Example: 00 08 01 02 03 04

Query message format for function code 08

	Slave Address (00 for TCP)	Function Code	Any data, length limited to approximately half the length of the data buffer	CRC (RTU)	CRC (RTU)
TCP Example	00	08	01 02 03 04		
RTU Example	01	08	01 02 03 04	CRC	CRC

Response

Example: 00 08 01 02 03 04

Response message format for function code 08

	Slave Address (00 for TCP)	Function Code	Data bytes received	CRC (RTU)	CRC (RTU)
TCP Example	00	08	01 02 03 04		
RTU Example	01	08	01 02 03 04	CRC	CRC

4.9 Function Codes 16 (10h) - Preset Multiple Registers

Description

Presets values into a sequence of holding registers (also refered to as 4X references). The HC900 does not support Broadcast. The register assignments specified in Section 6 with an access type "W", can be written to via Function Code 16 (10h).

Writing to half of a digital variable will be accepted. A non-zero write to either register will result in a floating point 1 to be written to the variable. Writing a zero to either register will result in the variable being set to 0.

Query

The query message specifies the register references to be preset. Registers are addressed starting at zero: Register 1 is addressed as 0.

Example: Preset Rack #1, Variable #1 (address 18C0h) to 100.0 in the controller at slave address 1.

Query message format for function code 16 (10h)

	Slave Address (00 for TCP)	Function Code	Start Address High	Start Address Low	Number Addresses High	Number Addresses Low	Byte Count	Data	CRC (RTU)	CRC (RTU)
TCP Example	00	10	18	C0	00	02	04	42 C8 00 00		
RTU Example	01	10	18	C0	00	02	04	42 C8 00 00	CRC	CRC

Response

The normal response returns the slave address, function code, starting address and the quantity of registers preset.

The floating point values require two consecutive addresses. A request to preset a single floating point value must be for two addresses. The byte order of the floating point number is determined by the setting of the byte swap configuration value. In this example the byte swap order is FP B. Refer to subsection 1.3. The first 16 bits of the response contain the IEEE MSB of the float value. The second 16 bits of the response contain the IEEE LSB of the float value. The Byte order is configurable See Subsection 1.3. If the master station requests only one address at an address of a floating point value the slave will respond with an illegal data address exception code (See Section 5).

Example: Response from presetting Rack #1, Variable #1 (address 18C0h) to 100.0 from the controller.

Response message format for function code 16 (10h)

	Slave Address (00 for TCP)	Function Code	Start Address High	Start Address Low	Number Addresses High	Number Addresses Low	CRC (RTU)	CRC (RTU)
TCP Example	00	10	18	C0	00	02		
RTU Example	01	10	18	C0	00	02	CRC	CRC

4.10 Function Code 17 (11h) - Report HC900 ID

Description

Function code 17 (11h) is used to report the device information such as Slave ID, device description and firmware version.

Query

The query message specifies the function code only.

Example: Read Device ID from a slave at address 2.

Query message format for function code 17 (11h)

	Slave Address (00 for TCP)	Function Code	CRC (RTU)	CRC (RTU)
TCP Example	00	11		
RTU Example	02	11	CRC	CRC

Response

The response is a record format describing the instrument.

Response message format for function code 17 (11h)

Slave Address	Function Code	Byte Count	Slave ID	Run Indicator Status	Device Description	Model ID	Device Class ID	Device Mapping	CRC (RTU)	CRC (RTU)

Slave Address – 00 for TCP. RTU is slave address. Example: 02

Slave ID - The Slave ID number for the HC930 is 93 (hex). For the HC950 it is 95 (hex) (one byte) (byte 3). For HC970 it is 70 (hex). For HC970R it is 7A (hex).

Run Indicator Status - (one byte) (byte 4). 00=OFF; FF=ON

Device Description - (bytes 5-20)- 16 Character ASCII Message with the following format:

'H'	'C'	'9'	'5' or '3' or '7'	'0' or 'A'	''	"	*up to 9 character version number in floating point notation.*	*zeros are appended for the remaining bytes*

For example, an HC950 with version number 2.000 would have the following device description:

'H'	'C'	'9'	'5'	'0'	''	"	*'2'*	*'.'*	*'0'*	*'0'*	*'0'*	*0*	*0*	*0*	*0*

Model ID - 00 (one byte) (byte 21)

Device Class ID - The Device Classification. (one byte) (byte 22)

Class ID	Class
00	Generic Class (Fixed Address Mappable)
01-FF	Future

Continued

Generic Class (00) Device Mapping - Describes the I/O and feature mapping.

Number of Records	Record #1	Record #2	Record ...	Record #n

Number of Records - 1 Byte unsigned value 00-FFh (byte 23)

Record Description:

Byte	Description
00	Type of Data Element (See Data Element Values Table Below)
01	Starting Address of Data Element Record (High)
02	Starting Address of Data Element Record (Low)
03	Number of Data Elements (High)
04	Number of Data Elements (Low)

Data Element Values Table:

Value	Description
00	Analog Inputs
01	Not Applicable. Number of elements equals 0.
02	Discrete Inputs
03	Discrete Outputs
04	Control Loops
05	Set Point Programmers
06	Variables
07	Not applicable. Number of elements equals 0.
08	Not applicable. Number of elements equals 0.
09	Not applicable. Number of elements equals 0.
10	Schedulers
11	Tagged Signals
12	Tagged Signals List 2
13	Analog Inputs Accessed Using 3x Registers
14	Sequencers
15	Stage Groups
16	Ramp Groups
17	Hand Off Auto
18	Alternators
19	Device Controls
20	Control Loops Extended
21	Setpoint Programmers Extended
22	Schedulers Extended
23	Sequencers Extended

5. Modbus RTU Exception Codes

5.1 Introduction

When a master device sends a query to a slave device it expects a normal response. One of four possible events can occur from the master's query:

- *Slave device receives the query without a communication error and can handle the query normally.* It returns a normal response.
- *Slave does not receive the query due to a communication error.* No response is returned. The master program will eventually process a time-out condition for the query.
- *Slave receives the query but detects a communication error (parity, LRC or CRC).* No response is returned. The master program will eventually process a time-out condition for the query.
- *Slave receives the query without a communication error but cannot handle it (i.e., request is to a non-existent coil or register).* The slave will return with an exception response informing the master of the nature of the error (Illegal Data Address.)

The exception response message has two fields that differentiate it from a normal response:

Function Code Field:
In a normal response, the slave echoes the function code of the original query in the function code field of the response. All function codes have a most-significant bit (MSB) of 0 (their values are below 80 hex). In an exception response, the slave sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hex higher than the value would be for a normal response.

With the function code's MSB set, the master's application program can recognize the exception response and can examine the data field for the exception code.

Data Field:
In a normal response, the slave may return data or statistics in the data field. In an exception response, the slave returns an exception code in the data field. This defines the slave condition that caused the exception.

Query

Example: Internal slave error reading 2 registers starting at address 1820h from the controller at slave address 01.

TCP: 00 03 18 20 00 02
RTU: 01 03 18 20 00 02 CRC CRC

Response

Example: Return MSB in Function Code byte set with Slave Device Failure (04) in the data field.

TCP: 00 83 04
RTU: 01 83 04 CRC CRC

Table 5-1 Modbus RTU Data Layer Status Exception Codes

Exception Code	Definition	Description
01	Illegal Function	The message received is not an allowable action for the addressed device.
02	Illegal Data Address	The address referenced in the function-dependent data section of the message is not valid in the addressed device.
03	Illegal Data Value	The value referenced at the addressed device location is not within range.
04	Slave Device Failure	The addressed device has not been able to process a valid message due to a bad device state.
06	Slave Device Busy	The addressed device has rejected a message due to a busy state. Retry later.
07	NAK, Negative Acknowledge	The addressed device cannot process the current message. Issue a PROGRAM POLL to obtain device-dependent error data.

6. Register Map for Process and Operation Type Variables

What's in this section?

This section describes all parameters accessible by Function Code 03, 04, 06 and 10h. Section 6.1 gives a global overview of each function and its addresses/registers. Sections 6.2 through 6.28 contain the details on each function and each of its parameters.

Your particular controller may not contain all parameters shown. If you see a function that is not on your controller, either it is not available for that controller model or it is not in your configuration.

6.1 Register Map Overview

Table 6-1 describes the global register map for Function Code 03, 04, 06 and 10h. Details on each address are in sections 6.2 through 6.28.

Conversion between Address (Hex) Number and Register (Decimal) Number
To convert the address number to the register number, convert the address from hexidecimal to decimal and add 1. Registers are addressed starting at zero: registers 1-16 are addressed as 0-15.

To convert the register number to the address number, subtract 1 from the register and convert to hex.

Table 6-1 Global Register Map

Start Address (hex)	End Address (hex)	Description	See Subsection
0000	< 0040	Miscellaneous Parameters	6.2
0040	00FF	Loop #1 (floating point & bit packed)	6.3
0140	01FF	Loop #2 (floating point & bit packed)	
0240	02FF	Loop #3 (floating point & bit packed)	
0340	03FF	Loop #4 (floating point & bit packed)	
0440	04FF	Loop #5 (floating point & bit packed)	
0540	05FF	Loop #6 (floating point & bit packed)	
0640	06FF	Loop #7 (floating point & bit packed)	
0740	07FF	Loop #8 (floating point & bit packed)	
0840	08FF	Loop #9 (floating point & bit packed)	
0940	09FF	Loop #10 (floating point & bit packed)	
0A40	0AFF	Loop #11 (floating point & bit packed)	
0B40	0BFF	Loop #12 (floating point & bit packed)	
0C40	0CFF	Loop #13 (floating point & bit packed)	
0D40	0DFF	Loop #14 (floating point & bit packed)	
0E40	0EFF	Loop #15 (floating point & bit packed)	
0F40	0FFF	Loop #16 (floating point & bit packed)	
1040	10FF	Loop #17 (floating point & bit packed)	
1140	11FF	Loop #18 (floating point & bit packed)	
1240	12FF	Loop #19 (floating point & bit packed)	
1340	13FF	Loop #20 (floating point & bit packed)	
1440	14FF	Loop #21 (floating point & bit packed)	
1540	15FF	Loop #22 (floating point & bit packed)	
1640	16FF	Loop #23 (floating point & bit packed)	
1740	17FF	Loop #24 (floating point & bit packed)	
7840	78FF	Loop #25 (floating point & bit packed)	

Start Address (hex)	End Address (hex)	Description	See Subsection
7940	79FF	Loop #26 (floating point & bit packed)	
7A40	7AFF	Loop #27 (floating point & bit packed)	
7B40	7BFF	Loop #28 (floating point & bit packed)	
7C40	7CFF	Loop #29 (floating point & bit packed)	
7D40	7DFF	Loop #30 (floating point & bit packed)	
7E40	7EFF	Loop #31 (floating point & bit packed)	
7F40	7FFF	Loop #32 (floating point & bit packed)	
1800	187F	Analog Input Value (#1-#64) First 8 slots of Rack#1 - Function Code 03	6.5
		Analog Input Value - Function Code 04	6.6
18C0	1D6F	Variable Value (#1 - #600)	6.7
1DF0	1DF6	Time	6.8
1E00	1E0F	Set Point Programmer #1	6.10
1E10	1E1F	Set Point Programmer #2	
1E20	1E2F	Set Point Programmer #3	
1E30	1E3F	Set Point Programmer #4	
8000	800F	Set Point Programmer #5	
8010	801F	Set Point Programmer #6	
8020	802F	Set Point Programmer #7	
8030	803F	Set Point Programmer #8	
1F00	1F3F	Set Point Programmer #1 Additional Values	6.12
1F40	1F7F	Set Point Programmer #2 Additional Values	
1F80	1FBF	Set Point Programmer #3 Additional Values	
1FC0	1FFF	Set Point Programmer #4 Additional Values	
8070	80AF	Set Point Programmer #5 Additional Values	
80B0	80EF	Set Point Programmer #6 Additional Values	
80F0	812F	Set Point Programmer #7 Additional Values	
8130	816F	Set Point Programmer #8 Additional Values	
2000	27CF	Tagged Signal (#1 - #1000)	6.9
3B60	5A9F	Tagged Signal (#1 - #4000)	
2800	29FF	Set Point Programmer #1 Segments	6.13
2A00	2BFF	Set Point Programmer #2 Segments	
2C00	2DFF	Set Point Programmer #3 Segments	
2E00	2FFF	Set Point Programmer #4 Segments	
8200	83FF	Set Point Programmer #5 Segments	
8400	85FF	Set Point Programmer #6 Segments	
8600	87FF	Set Point Programmer #7 Segments	
8800	89FF	Set Point Programmer #8 Segments	
3000	304F	Scheduler #1 Value	6.15
3050	309F	Scheduler #2 Value	
3200	3B5F	Scheduler #1 Segments	6.16
6C00	755F	Scheduler #2 Segments	
5AA0	5ABF	Sequencer #1 Parameters	6.19

Start Address (hex)	End Address (hex)	Description	See Subsection
5AC0	5CBF	Sequencer #1 Step 1 Table (64 steps in sequence)	6.20
5CC0	5CFF	Sequencer #1 State Table	6.21
5D00	5D1F	Sequencer #2 Parameters	6.19
5D20	5F1F	Sequencer #2 Step 1 Table (64 steps in sequence)	6.20
5F20	5F5F	Sequencer #2 State Table	6.21
5F60	5F7F	Sequencer #3 Parameters	6.19
5F80	617F	Sequencer #3 Step 1 Table (64 steps in sequence)	6.20
6180	61BF	Sequencer #3 State Table	6.21
A000	A01F	Sequencer #4 Parameters	6.19
A020	A21F	Sequencer #4 Step 1 Table (64 steps in sequence)	6.20
A220	A25F	Sequencer #4 State Table	6.21
6600	6606	Hand Off Auto #1	6.22
6610	6616	Hand Off Auto #2	
6620	6626	Hand Off Auto #3	
6630	6636	Hand Off Auto #4	
6640	6646	Hand Off Auto #5	
6650	6656	Hand Off Auto #6	
6660	6666	Hand Off Auto #7	
6670	6676	Hand Off Auto #8	
6680	6686	Hand Off Auto #9	
6690	6696	Hand Off Auto #10	
66A0	66A6	Hand Off Auto #11	
66B0	66B6	Hand Off Auto #12	
66C0	66C6	Hand Off Auto #13	
66D0	66D6	Hand Off Auto #14	
66E0	66E6	Hand Off Auto #15	
66F0	66F6	Hand Off Auto #16	
6B00	6B09	Device Control #1	6.23
6B10	6B19	Device Control #2	
6B20	6B29	Device Control #3	
6B30	6B39	Device Control #4	
6B40	6B49	Device Control #5	
6B50	6B59	Device Control #6	
6B60	6B69	Device Control #7	
6B70	6B79	Device Control #8	
6B80	6B89	Device Control #9	
6B90	6B99	Device Control #10	
6BA0	6BA9	Device Control #11	
6BB0	6BB9	Device Control #12	

Start Address (hex)	End Address (hex)	Description	See Subsection
6BC0	6BC9	Device Control #13	6.23
6BD0	6BD9	Device Control #14	
6BE0	6BE9	Device Control #15	
6BF0	6BF9	Device Control #16	
6700	6725	Alternator #1	6.24
6730	6755	Alternator #2	
6760	6785	Alternator #3	
6790	67B5	Alternator #4	
67C0	67E5	Alternator #5	
67F0	6815	Alternator #6	
6820	6845	Alternator #7	
6850	6875	Alternator #8	
6880	68A5	Alternator #9	
68B0	68D5	Alternator #10	
68E0	6905	Alternator #11	
6910	6935	Alternator #12	
6940	6965	Alternator #13	
6970	6995	Alternator #14	
69A0	69C5	Alternator #15	
69D0	69F5	Alternator #16	
6A00	6A10	Output Order Sequence Scrarch Pad	6.25
6200	6227	Stage Group #1	6.26
6230	6257	Stage Group #2	
6260	6287	Stage Group #3	
6290	62B7	Stage Group #4	
62C0	62E7	Stage Group #5	
62F0	6317	Stage Group #6	
6320	6347	Stage Group #7	
6350	6377	Stage Group #8	
6400	642D	Ramp Group #1	6.27
6430	645D	Ramp Group #2	
6460	648D	Ramp Group #3	
6490	64BD	Ramp Group #4	
64C0	64ED	Ramp Group #5	
64F0	651D	Ramp Group #6	
6520	654D	Ramp Group #7	
6550	657D	Ramp Group #8	
B000	B3E7	User defined signal or variable. Read-only if a signal tag. Read/write if a variable.	6.28

6.2 Miscellaneous Register Map

Table 6-2 Miscellaneous Register Map Addresses

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
0000	0001	Instrument Mode	R/W	Bit Packed Indicators: Bit 0: 1:Diagnostic Bit 1: 1:Calibration (unused by UMC800/HC9xx) Bit 2: 1:Maintenance/Offline mode Bit 3: 1:Program mode Bit 4: 1:Reset Unit/Force Cold Start (Write Only) Bit 5: 1:On-Line/Run mode Bit 6: 1:Fail-over (Write Only; C70R Only) Bit 7: Lead slot position: 0: A; 1: B (C70R Only) Bit 8: 1:On-demand write database to FLASH Bit 9: Database write status: 0: Done; 1: Busy Bit 10-15: Unused
0002	0003	Load Recipe	R/W	Floating Point Read: Returns zero Write: Loads the recipe, identified in the request, from the recipe pool.
0004	0005	Reserve Status	R	Bit Packed Bit 0: Available for failover Bit 1: Available for failover with no RSM Bit 2: Available for failover with switch bad Bit 3: Available for failover with scanner communication failure Bit 4: Unsynchronized database Bit 5: Invalid database Bit 6: Software version mismatch Bit 7-15: Reserved *C70R Only*

6.3 Loop Value Register Map

This table contains addresses of Loop #1. ***See the Global Register Map - Table 6-1 for starting and ending addresses (hex) for Loop #2 through Loop #32***. Each successive control loop is offset by 256 with the exception that loop 25 has a new starting address and loop 26 - 32are offset by 256. The loop number corresponds to the PID block entry sequence during Hybrid Control Designer configuration. The Modbus loop number address for a loop can also be obtained from the Hybrid Control Designer printout of Block Modbus Addresses.

Function Code Support:

Reads – Function Code 3

Writes – Function Code 16 (10 hex) for preset of multiple registers (e.g., for floating point)

Writes – Function Code 6 for presetting an integer value

Table 6-3 Loop Value Register Map Addresses

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
0040	0065	PV	R	Floating Point in Engineering Units.
0042	0067	Remote Set Point; SP2	R/W	Floating Point in Engineering Units. When the remote setpoint source is configured as LSP2, the value can be written.
0044	0069	Working Set Point	R/W	Floating Point in Engineering Units. On a write to this register the instrument will update the proper set point according to the loop's currently selected set point.
0046	0071	Output	R/W	Floating Point in Engineering Units.
0048	0073	PV	R	Floating Point in Engineering Units.
004A	0075	Carbon Potential block temperature	R	Floating Point in Engineering Units
004C	0077	Gain #1 (Prop Band #1 *if active*)	R/W	Floating Point . (in units per what was configured in the HC900: Gain or Proportional Band)
004E	0079	Direction	R	Floating Point 0.0=Direct; 1.0=Reverse
0050	0081	Reset #1	R/W	Floating Point in Repeats/Minute or Minutes/Repeat.
0052	0083	Rate #1	R/W	Floating Point in Minutes
0054	0085	Cycle Time for Analog Scan	R	Floating Point in Seconds
0056	0087	PV Low Range	R	Floating Point in Engineering Units.
0058	0089	PV High Range	R	Floating Point in Engineering Units.
005A	0091	Alarm #1 SP #1	R/W	Floating Point in Engineering Units.
005C	0093	Alarm #1 SP #2	R/W	Floating Point in Engineering Units.

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
0060	0097	Gain #2 (Prop Band #2 *if active*)	R/W	Floating Point
0062	0099	Three Position Step Motor Deadband	R/W	Floating Point in percent
0064	0101	Reset #2	R/W	Floating Point in Repeats/Minute or Minutes/Repeat as configured in the HC900.
0066	0103	Rate #2	R/W	Floating Point in Minutes
0068	0105	Cycle Time for Analog Scan	R	Floating Point in Seconds
006A	0107	LSP #1	R/W	Floating Point in Engineering Units
006C	0109	LSP #2	R/W	Floating Point in Engineering Units.
006E	0111	Alarm #2 SP #1	R/W	Floating Point in Engineering Units.
0070	0113	Alarm #2 SP #2	R/W	Floating Point in Engineering Units.
0074	0117	SP Low Limit	R/W	Floating Point in Engineering Units. Operator Limit
0076	0119	SP High Limit	R/W	Floating Point in Engineering Units. Operator Limit
0078	0121	Working Set Point	R/W	Floating Point in Engineering Units. On a write to this register the instrument will update the proper set point according to the loop's currently selected set point.
007A	0123	Output Low Limit	R/W	Floating Point in Engineering Units.
007C	0125	Output High Limit	R/W	Floating Point in Engineering Units.
007E	0127	Output Working Value	R/W	Floating Point in Engineering Units.
0086	0135	Ratio	R/W	Floating Point in Engineering Units.
0088	0137	Bias	R/W	Floating Point in Engineering Units. Auto/Man bias block, value is Read-Only
008A	0139	Deviation	R	Floating Point in Engineering Units. (SP-PV)
008E	0143	Manual Reset	R/W	Floating Point in Engineering Units.
0090	0145	Feed-forward Gain	R/W	Floating Point in Engineering Units.
0092	0147	Local Percent Carbon Monoxide	R/W	Floating Point in Engineering Units.
0094	0149	Furnace Factor	R/W	Floating Point in Engineering Units.
0096	0151	Percent Hydrogen	R/W	Floating Point in Engineering Units.
0098	0153	On/Off Output Hysterisis	R/W	Floating Point in percent of input span
009A	0155	Carbon Potential Dewpoint	R/W	Floating Point in Engineering Units
009C	0157	Three Position Step Motor Time	R/W	Floating Point in seconds

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
00F7	0248	Enable/Disable Fuzzy	R/W	Bit Packed Bit 0: 0:Disable; 1:Enable
00F8	0249	Demand Tune Request	R/W	Bit Packed (one shot action, activates autotuning until autotuning completed) Bit 0: 0:Off; 1:On Bit 1-15: Unused
00F9	0250	Anti-soot set point limit enable	R/W	Bit Packed Bit 0: 0:Off; 1:On Bit 1-15: Unused
00FA	0251	Auto/Manual State	R/W	Bit Packed Bit 0: 0:Manual; 1:Auto Bit 1-15: Unused
00FB	0252	Set Point State	R/W	Bit Packed (selects either SP1 or SP2 as a local SP if the HC900 is configured as such in PID setup) Bit 0: 0:SP1; 1:SP2 Bit 1-15: Unused
00FC	0253	Remote/Local Set Point State	R/W	Bit Packed Bit 0: 0:LSP; 1:RSP Bit 1-15: Unused
00FD	0254	Tune Set State	R/W	Bit Packed (selects tuning constant set) Bit 0: 0:Tune Set #1; 1:Tune Set #2 Bit 1-15: Unused
00FE	0255	Loop Status	R	Bit Packed Bit 0: Mode: 0:Manual; 1:Auto Bit 1: Set Point: 0:SP1; 1:SP2 Bit 2: Remote/Local: 0:LSP; 1:RSP Bit 3: Tune Set: 0:Set #1; 1:Set #2 Bit 4: IMAN: 0:Inactive; 1:Active Bit 5: LO: 0: Inactive; 1:Active Bit 6-15: Reserved

6.4 Example for queries using Function Codes 3, 6, 16

Example 1

Query: Read PV, Remote SP, Working SP, Output for Loop 1 from HC900 at slave address 01 using Function Code 3 (hex codes). This will be accomplished by accessing contiguous registers.

TCP:

00	03	00	40	00	08

RTU:

01	03	00	40	00	08	CRC	CRC

Response: where PV=1000.0, Remote SP=1000.0, Working SP=1000.0, Output=50.0

TCP:

00	03	10	44	7A	00	00	44	7A	00	00	44	7A	00	00	42	48	00	00

RTU:

01	03	10	44	7A	00	00	44	7A	00	00	44	7A	00	00	42	48	00	00	CRC	CRC

Example 2:

Query: Write a Local Setpoint , (address 006A) to 100.0 for loop 1 at HC900 at slave address 01 using Function Code 16 (10 hex). Function code 16 is used for presetting multiple registers. Registers are addressed starting at zero: Register 1 is addressed as 0.

TCP:

00	10	00	6A	00	02	04	42	C8	00	00

RTU:

01	10	00	6A	00	02	04	42	C8	00	00	CRC	CRC

Response: from preset of LSP#1, address 006A to 100.0.

TCP:

00	10	00	6A	00	02

RTU:

01	10	00	6A	00	02	CRC	CRC

Example 3:

Query: Using Function code 6, set Loop #1 to Auto (register 00FA hex) at controller at slave address 01. This is a bit-packed single register data type. Registers are addressed starting at 0: Register 1 is addressed as 0.

TCP:

00	06	00	FA	00	01

RTU:

01	06	00	FA	00	01	CRC	CRC

Response: from preset of LSP#1, address 006A to 100.0 at address 1.

TCP:

00	06	00	FA	00	01

RTU:

01	06	00	FA	00	01	CRC	CRC

6.5 Analog Input Value Register Map - Function Code 03

Summary

Used to access analog input parameters for the first 8 channels of the first 8 modules of Rack #1. (Address compatible with UMC800.)

To access more analog input channels use Function Code 04.

Analog Input Example: AI1 through AI64. The mapping is with respect to card position starting with the first card module position and continuing to the 8th module position. The first module position would be AI1 thru AI8. The next AI card in slot 2 would be AI9 thru AI16 and so on.

Function Code Support:

Reads – **Function Codes 03 (limited to the first 8 slots of Rack #1)**

Table 6-4 Analog Input Value Register Map Addresses - Function Code 03

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1800	6145	Analog Input #1	R	Floating Point in Engineering Units.
1802	6147	Analog Input #2	R	
1804	6149	Analog Input #3	R	
1806	6151	Analog Input #4	R	
1808	6153	Analog Input #5	R	
180A	6155	Analog Input #6	R	
180C	6157	Analog Input #7	R	
180E	6159	Analog Input #8	R	
1810	6161	Analog Input #9	R	
1812	6163	Analog Input #10	R	
1814	6165	Analog Input #11	R	
1816	6167	Analog Input #12	R	
:	:	:		
187E	6271	Analog Input #64	R	

Example

Read Analog Inputs 1 and 2 from HC900 at slave address 01 using Function Code 03.

TCP:

00	03	18	00	00	04

RTU:

01	06	00	FA	00	01	CRC	CRC

Response from HC900 where AI1 = 100.0 and AI 2 = 55.32

TCP:

00	03	08	42	C8	00	00	42	5D	47	AE

RTU:

01	03	08	42	C8	00	00	42	5D	47	AE	CRC	CRC

6.6 Analog Input Value Register Map - Function Code 04

Summary

Used to access analog input parameters positioned in any Rack or Slot. Refer to section 4.5 for the address mapping for HC900 firmware release 2.3 or 2.4.

6.7 Variable Register Map

Summary

Variables (analog or digital) are writeable parameters in HC900 attached to input pins of function blocks. Digital Variable status is also represented in floating point: 0.0 for OFF or logic 0 and 1.0 for ON or logic 1. The Variable number in the table corresponds to the Variable number in the HC900 configuration.

You will need to access the HC900 Hybrid Control Designer configuration or corresponding configuration printout of Tag Information Report to identify the Variable numbers desired.

Single byte writes are permitted to digital variables. A non-zero will automatically set the digital variable to a floating point 1 and a zero write to a single register will set the digital variable to 0.

Function Code Support:

Read – Function Code 3

Write – Function Codes 6 (digital variables only), 16 (10 Hex)

Table 6-5 Variable Register Map Addresses

Address (hex)	Register (decimal)	Channel Number	Access	Notes
18C0	6337	Variable Value #1	R/W	Floating Point in Engineering Units.
18C2	6339	Variable Value #2	R/W	
18C4	6341	Variable Value #3	R/W	
18C6	6343	Variable Value #4	R/W	:
18C8	6345	Variable Value #5	R/W	:
18CA	6347	Variable Value #6	R/W	:
18CC	6349	Variable Value #7	R/W	:
18CE	6351	Variable Value #8	R/W	:
18D0	6353	Variable Value #9	R/W	:
18D2	6355	Variable Value #10	R/W	:
18D4	6357	Variable Value #11	R/W	:
18D6	6359	Variable Value #12	R/W	:
18D8	6361	Variable Value #13	R/W	:
18DA	6363	Variable Value #14	R/W	:
18DC	6365	Variable Value #15	R/W	:
18DE	6367	Variable Value #16	R/W	:
18E0	6369	Variable Value #17	R/W	:
18E2	6371	Variable Value #18	R/W	:
18E4	6373	Variable Value #19	R/W	:
18E6	6375	Variable Value #20	R/W	:
:	:	:		:
1D6E	7535	Variable Value #600	R/W	:

continued

Example

Query: Read Variables 1 and 2 from HC900 at address 1 using Function Code 3 (hex codes).

TCP:

00	03	18	C0	00	04

RTU:

01	03	18	C0	00	04	CRC	CRC

Response: from HC900 where Variable 1 = 100.0 and Variable 2 = 55.32

TCP:

00	03	08	42	C8	00	00	42	5D	47	AE

RTU:

01	03	08	42	C8	00	00	42	5D	47	AE	CRC	CRC

6.8 Time Register Map

Table 6-6 Time Register Map Addresses

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1DF0	7665	Hours	R/W	0 to 23
1DF1	7666	Minutes	R/W	0 to 60
1DF2	7667	Seconds	R/W	0 to 60
1DF3	7668	Month	R/W	1 to 12
1DF4	7669	Day	R/W	1 to 31
1DF5	7670	Year	R/W	1970 to 2037 *The values read are always in the range of 1970 to 2037 for a write. 0 – 37 represents 2000 – 2037, 70 – 99 represents 1970 – 1999*
1DF6	7671	Week Day	R	0 to 6 (0 = Sunday)

ATTENTION

Clock registers must be written in a single transaction. They can be written in one transaction of registers 7665 through 7670 or one transaction of registers 7665 through 7671.

6.9 Signal Tag Register Map

Summary

Signal tags are connected to output pins of function blocks, representing analog or digital parameters, and are read-only parameters. Digital Signal tags are also represented in floating point, 0.0 for OFF or logic 0, 1.0 for ON or logic 1. The signal tag number in the table corresponds to the signal tag number in the HC900 Hybrid Control Designer configuration. You will need to access the Hybrid Control Designer configuration "Tag Information" report to identify the Signal Tag numbers desired.

Note: to convert floating point values (analog or digital) to integer 16, for use with third party touch panels and associated HMI software, for example, requires the use of the *user-defined* modbus map for assigning new modbus addresses and associated data type, configurable in HC Designer, Ver. 2.0 or later.

Function Code Support:

Read – Function Code 3

NOTES:

Floating Point in Engineering Units

Digital Signal Tags are represented as 0.0 for OFF, 1.0 for ON.

Access is Read Only

Table 6-7 Signal Tag Register Map Addresses

Legacy			HC900 Range		
Address (hex)	**Register (decimal)**	**Channel Number**	**Address (hex)**	**Register (decimal)**	**Channel Number**
2000	8193	Tagged Signal #1	3B60	15201	Tagged Signal #1
2002	8195	Tagged Signal #2	3B62	15203	Tagged Signal #2
2004	8197	Tagged Signal #3	3B64	15205	Tagged Signal #3
2006	8199	Tagged Signal #4	3B66	15207	Tagged Signal #4
2008	8201	Tagged Signal #5	3B68	15209	Tagged Signal #5
200A	8203	Tagged Signal #6	3B6A	15211	Tagged Signal #6
200C	8205	Tagged Signal #7	3B6C	15213	Tagged Signal #7
200E	8207	Tagged Signal #8	3B6E	15215	Tagged Signal #8
2010	8209	Tagged Signal #9	3B70	15217	Tagged Signal #9
2012	8211	Tagged Signal #10	3B72	15219	Tagged Signal #10
. . . 27CF	. . . 10192	. . . Tagged Signal #1000	42CB	17099	. . . Tagged Signal #1000
			42CD 5A9F	17101 23200	Tagged Signal #1001 . . . Tagged Signal #4000

Example

Query: Read Signal Tags 3 and 4 from HC900 at address 1 using Function Code 3 (hex codes).

TCP:

00	03	20	04	00	04

RTU:

01	03	20	04	00	04	CRC	CRC

Response: from HC900 where Signal Tag 3 = 100.0 and Signal Tag 4 = 55.32

TCP:

00	03	08	42	C8	00	00	42	5D	47	AE

RTU:

01	03	08	42	C8	00	00	42	5D	47	AE	CRC	CRC

6.10 Set Point Program Register Maps

Summary

The SP Programmer parameters are listed according to category related to program status and interaction, other programmer parameters and program segment mapping. A section is also provided to aid in configuring a SP programmer and recipe interface for third party software.

Function Code Support:

Read – Function Code 3

Write – Function Code 16 (10 Hex)

Considerations and Methods for Downloading. Operating, and Reading Status of SP Programs:

A SP programmer interface can be developed (and recipes containing a SP profile can be created) in third party software using the parameters listed in the following table.

In creating a SP Programmer interface showing a number of segments, a graphic display might include a table referencing the maximum number of ramp/soak segments that you will be using for your process. The parameters to be referenced for each segment are listed in Table 6-14 - Register Map (Ramp/Soak Segments).

Programmer Numbers

The parameters that follow refer to SP Programmer 1. Table 6-8 lists the Starting and Ending addresses for all of the SP Programmers.

Table 6-8 SP Programmer Addresses

Programmer Number	Starting Address (Hex)	Starting Register (Decimal)	Ending Address (Hex)	Ending Register (Decimal)
SP Programmer 1	1E00	7681	1E0F	7696
SP Programmer 2	1E10	7697	1E1F	7712
SP Programmer 3	1E20	7713	1E2F	7728
SP Programmer 4	1E30	7729	1E3F	7744
SP Programmer 5	8000	32769	800F	32784
SP Programmer 6	8010	37785	801F	32800
SP Programmer 7	8020	32801	802F	32816
SP Programmer 8	8030	32817	803F	32832

Controlling the Program

For controlling the program, the following parameters should be accessed:

Parameter	Addr (Hex)	Register (Decimal)	Notes
SP Programmer Output	1E00	7681	See Table 6-11
Current Segment Number	1E02	7683	See Table 6-11
Program Elapsed Time	1E04	7685	See Table 6-11
Segment Time Remaining	1E08	7689	See Table 6-11
Current Segment Events	1E0A	7691	See Table 6-11
Status	1E0B	7692	See Table 6-11
Start (write only)	1E0C	7693	See Table 6-11
Hold (write only)	1E0D	7694	See Table 6-11
Advance (write only)	1E0E	7695	See Table 6-11
Reset (write only)	1E0F	7696	See Table 6-11
Current Program Number	1F00	7937	See Table 6-12
Auxiliary Output	1F04	7941	See Table 6-12

Parameters for the Profile

You will also need to define the following for the parameters for the profile:

Parameter	Addr (Hex)	Register (Decimal)	Notes
Time Units	1F3A	7995	See Table 6-12
Ramp Units (segments)	1F3B	7996	See Table 6-12
Guaranteed Soak Type	1F3C	7997	See Table 6-12
Guaranteed Soak Low	1F06	7943	See Table 6-12
Guaranteed Soak High	1F08	7945	See Table 6-12
Prog Save Request	1F02	7939	See Table 6-12

Procedures for Downloading Setpoint Programs

These steps are for programmer 1. For programmers 2, 3, 4, 5, 6, 7, and 8 adjust the register addresses accordingly by adding offset for starting addresses.

Table 6-9 is for downloading using Function Codes 3, 4, 6, 16.

Table 6-9 Steps to Download a Setpoint Program using Modbus Function Codes 3, 6, 16

Step	Action
1	Set the programmer to RESET by writing any number to 7696 (1E0F). This can be done either with function code 6 or 16.
2	Clear the program by writing a 0 to registers 7937 and 7938 (1F00 and 1F01). This is a floating point register and requires a multiple register write (function code 16). This is the safest way to insure that all registers are cleared for the next program download.
3	Write the header information for parameters relevant to the program (leave all others at 0) -- registers 7943 (1F06) - 7997 (1F3C). Registers 7943 - 7968 are floats and must be written using function code 16. Registers 7995 - 7997 are bit-packed and can be written with either function code 6 or 16. Note: Display High Range Limit and Display Low Range Limit are not presently used in the HC900.
4	Write the information for each segment required in the profile -- registers 2800 - 2807 for segment 1, 2808 - 280F for segment 2, etc. The first 2 registers are bit-packed and can be written with either function code 6 or 10. The rest of the registers are float and must be written using function code 16.
5	Save the program to a program (profile) number archive by writing a floating point number to register 7939. This will store the downloaded data utilized by the programmer block to the program (profile) number used. Profile numbers may range from 1-99.
The program is now ready to run. Note that the current program (profile) number -- register 7937 -- is automatically set to the saved program number.	

Procedure for Uploading Setpoint Programs

Table 6-10 Steps to Upload a Setpoint Program using Modbus Function Codes 3, 6, 16

Step	Action
1	Set the programmer to RESET by writing any number to 7696 (1E0F). This can be done either with function code 6 or 16.
2	Load the program into the setpoint programmer block by writing the program number to registers 7937 and 7938 (1F00 and 1F01). This is a floating point register and requires a multiple register write (function code 16).
3	Read the header information desired -- registers 7943 (1F06) - 7997 (1F3C) using function code 3.
4	Read the information for each segment desired using function code 3 — registers 2800 - 2807 for segment 1, 2808 - 280F for segment 2, etc.

6.11 Set Point Programmer Value Register Map

This table contains Value Register addresses of SP Programmer #1. ***See the Global Register Map - Table 6-1 for starting and ending addresses (hex) for SP Programmer #2 through SP Programmer #8 Value Register Map Addresses.***

Table 6-11 Set Point Programmer #1 Value Register Map Addresses

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1E00	7681	Set Point Programmer Output	R	Floating Point in Engineering Units.
1E02	7683	Current Segment Number	R/W	Floating Point; 1...Max Segment # 50 A write changes the segment number.
1E04	7685	Program Elapsed Time	R	Floating Point in Minutes Continues to run when in Hold
1E08	7689	Segment Time Remaining	R	Floating Point in Minutes
1E0A	7691	Current Segment Events	R	Bit Packed Indicates status of events 1-16 in one register Bit 0: Event #1 : Bit 15: Event #16 0: Event OFF 1: Event ON
1E0B	7692	Status	R	Bit Packed Bit 0: 1=Ready 1: 1=Run 2: 1=Hold 3: 1=End 4: 1=Reserved 5: 1=Time Units in Minutes 6: 1=Time Units in Hours 7: Ramp Units 0: Time 1: Rate 8: Reserved 9: If bit 2 Set 0: Operator hold 1: Guaranteed soak hold 10: 0: Current segment is a soak 1: Current segment is a ramp 11-15: Reserved
1E0C	7693	Run	W	Signed 16 bit integer Write to location Starts Profile; Data ignored
1E0D	7694	Hold	W	Signed 16 bit integer Write to location Holds Profile; Data ignored

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1E0E	7695	Advance	W	Signed 16 bit integer Write to location Advances Profile one segment while in Hold mode; Data ignored
1E0F	7696	Reset	W	Signed 16 bit integer Write to location Resets Profile after program is first in Hold mode; Data ignored

6.12 Set Point Programmer Additional Values Register Map

This table contains Additional Value Register addresses of SP Programmer #1. ***See the Global Register Map - Table 6-1 for starting and ending addresses (hex) for SP Programmer #2 through SP Programmer #8 Additional Value Register Map Addresses.***

Table 6-12 Set Point Programmer #1 Additional Values Register Map Addresses

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1F00	7937	Current Program Number	R/W	Floating Point Indicates the present profile number in use. This also allows entry of a profile number from the HC900 stored profile memory (if profiles have been stored in the controller) and will retrieve the profile data for a display showing a SP profile segment table. Typically, when recipes are downloaded from third party software, this will be the number generated by the Program Save Request parameter A write to this register loads the program into the set point programmer function block; if 0 is written, the function block's program is cleared.
1F02	7939	Program Save Request	R/W	Floating Point Assigns profile parameters downloaded to a profile number such as 1. **This is required by the HC900 to be the last parameter downloaded.** This overwrites what is in this HC900 SP Profile memory location on each download of a new SP profile. Saves the program into the archive. Writing to this register is prohibited in the run mode.
1F04	7941	Auxiliary Output	R	Floating Point
1F06	7943	Guaranteed Soak Low	R/W	Floating Point Presets High Deviation setting in engineering units. *Writing to this register is only permissible in the reset or ready mode*
1F08	7945	Guaranteed Soak High	R/W	Floating Point Presets Low Deviation setting in engineering units. *Writing to this register is only permissible in the reset or ready mode*
1F0A	7947	Restart Ramp Rate	R/W	Floating Point *Writing to this register is only permissible in the reset or ready mode*
1F0C	7949	Display High Range Limit	R/W	Floating Point *Writing to this register is only permissible in the reset or ready mode*

Address (hex)	Register (decimal)	Channel Number	Access	Notes
1F0E	7951	Display Low Range Limit	R/W	Floating Point *Writing to this register is only permissible in the reset or ready mode*
1F10	7953	Jog Segment	R/W	Floating Point *Writing to this register is only permissible in the reset or ready mode*
1F12	7955	Loop Start	R/W	Floating Point 0 indicates no loop. *Writing to this register is only permissible in the reset or ready mode*
1F14	7957	Loop End	R/W	Floating Point 0 indicates no loop. *Writing to this register is only permissible in the reset or ready mode*
1F16	7959	Repeats	R/W	Floating Point 0 indicates loop forever. *Writing to this register is only permissible in the reset or ready mode*
1F3A	7995	Time Units	R/W	Bit Packed Bit 0: Reserved 1: minutes 2: hours 3-15: Unused
1F3B	7996	Ramp Units	R/W	Bit Packed Bit 0: 0:Time; 1:Rate Bit 1-15: Unused *Writing to this register is only permissible in the reset or ready mode*
1F3C	7997	Guaranteed Soak Type	R/W	Bit Packed For selection of Soak (or Hold) guarantee – Per Segment (requiring selection per segment in a table) Bit 0: per segment 1: all soaks 2: all segments 3-15: Unused None if none of the bits is set *Writing to this register is only permissible in the reset or ready mode*

6.13 Set Point Programmer Segment Map

A profile contains up to 50 segments. Each segment is made up of 8 registers. This table contains Segment Map addresses of SP Programmer #1. ***See the Global Register Map - Table 6-1 for starting and ending addresses (hex) for SP Programmer #2 through SP Programmer #8 Segment Map Addresses.***

Table 6-13 Set Point Programmer #1 Segment Map Addresses

Start Address (Hex)	Start Register (Decimal)	End Address (Hex)	End Register (Decimal)	Description
2800	10241	2807	10248	Set Point Programmer #1 Segment 1
2808	10249	280F	10256	Set Point Programmer #1 Segment 2
2810	10257	2817	10264	Set Point Programmer #1 Segment 3
:		:		:
2988	10633	298F	10640	Set Point Programmer #1 Segment 50

6.14 Segment Register Map

The table below describes the registers that are part of a setpoint programmer segment. To determine the actual register address for a parameter within a segment, add the register offset to the start address of the segment.

Table 6-14 Segment Register Map Addresses

Register Offset within Segment	Parameter Name	Access	Notes
0	Ramp/Soak Segment Guaranteed Soak Enable	R/W	Bit Packed Bit 0: 1 = ramp segment; 0=soak segment Bit 1: 1 = guaranteed soak enabled 0 = guaranteed soak disabled Bit 0 is ignored in the hold mode. Writing to this register is not permissible in the run mode.
1	Events	R/W	Bit Packed Bit 0: Event #1 : : Bit 15: Event #16 0: Event OFF 1: Event ON Writing to this register is only permissible in reset or ready mode.
2	Time or Rate	R/W	Floating Point in time units configured for the set point programmer Writing to this register is not permissible in the run mode.
4	Ramp or Soak value	R/W	Floating Point Writing to this register is not permissible in the run mode.
6	Soak value for auxiliary output (use "Time or Rate" for duration)	R/W	Floating Point Writing to this register is not permissible in the run mode.

Example For Determining a Segment Register

To change the ramp value in segment #8 of setpoint programmer #2, the register address is determined as follows.

Step 1: Use Table 6-1 to determine the start address for setpoint program #2 profile. The value is 2A00 Hex.

Step 2: Calculate the offset address for segment 8 in a profile. This is calculated as:

Segment #8 offset address	=	(segment number – 1) * 8
	=	(8-1) * 8
	=	56 or 38 Hex

Step 3: Use the table above to determine the register offset for the ramp value. The value is 4.

Step 4: Calculate the address by adding the results of steps 1, 2, and 3 to determine the register address.

Register address	=	Setpoint program #2 profile base address + Segment 8 offset address + Ramp value register offset
	=	2A00 + 38 + 4
	=	2A3C

6.15 Scheduler Value Register Map

Summary

The SP Scheduler parameters are listed according to category related to SP Scheduler status plus interaction and scheduler segment mapping. A section is also provided to aid in configuring a Scheduler and recipe interface for third party software.

Function Code Support

Read – Function Code 3

Write – Function Code 16 (10 hex)

Scheduler Value Register Map

You will need to define the parameters for the Schedule as required by the application. Application notes for these parameters are provided and further defined in the **Scheduler Value Register Map**.

Table 6-18 contains Value Register addresses of SP Scheduler #1. See the Global Register Map - Table 6-1 for starting and ending addresses (hex) for SP Scheduler #2.

Scheduler Segment Register Map

Table 6-19 indicates the range of addresses applicable to a scheduler segment. Each segment uses 48 registers (30 hex).

Segment Register Map Addresses

Table 6-20 describes the registers that are part of a schedule segment. To determine the actual register address for a parameter within a segment, add the register offset to the start address of the segment.

Considerations and Methods for Downloading, Operating, and Reading Status of SP Schedules

A SP Scheduler interface can be developed (and recipes containing a schedule can be created) in third party software using the parameters listed in Table 6-20.

In creating a Scheduler interface showing a number of segments and outputs for each segment on a graphic display might include a table referencing the **maximum** number of segments that you will be using for your process. Refer to the **Scheduler Segment Register Map -** Table 6-19 for the range of addresses applied to each scheduler segment. Each segment uses 48 registers (30 hex). Use the **Segment Register Map -** Table 6-20 for the parameters to be referenced for read/write within each segment. Application notes for these segment parameters is provided.

Scheduler Numbers

The parameters that follow refer to SP Scheduler #1. Table 6-8 lists the Starting and Ending addresses for all of the SP Schedulers.

Table 6-15 SP Scheduler Addresses

Scheduler Number	Starting Address (Hex)	Starting Register (Decimal)	Ending Address (Hex)	Ending Register (decimal)
SP Scheduler 1	3000	12289	304F	12368
SP Scheduler 2	3050	12369	309F	12448

Procedures for Downloading Setpoint Schedules

Table 6-16 is for downloading using Function Codes 3, 6, 16.

Table 6-16 Steps to Download a Setpoint Schedule using Modbus Function Codes 3, 6, 16

Step	Action
1	Set the scheduler to RESET by writing any number to 12367 (304E). This can be done either with function code 6 or 16.
2	Clear the schedule by writing a 0 to registers 12321 and 12322 (3020 and 3021). This is a floating point register and requires a multiple register write (function code 16). This is the safest way to insure that all registers are cleared for the next schedule download.
3	Write the header information for parameters relevant to the schedule (leave all others at 0) -- registers 12331 (302A) - 12347 (303A) and register 12368 (304F). Registers 12331 - 12347 are floats and must be written using function code 16. Register 12368 is bit-packed and can be written with either function code 6 or 16.
4	Write the information for each segment required in the schedule -- registers 3200 – 322F for segment 1, 3230 – 325F for segment 2, etc. The first 9 registers are bit-packed and can be written with either function code 6 or 10. The rest of the registers are float and must be written using function code 16.
5	Save the schedule to a schedule number archive by writing a floating point number to register 12329. This will store the downloaded data utilized by the scheduler block to the schedule number used. Schedule numbers may range from 1-50.
The schedule is now ready to run. Note that the current schedule number -- register 12321-- is automatically set to the saved schedule number.	

Procedure for Uploading Setpoint Schedules

Table 6-17 Steps to Upload a Setpoint Schedule using Modbus Function Codes 3, 6, 16

Step	Action
1	Set the scheduler to RESET by writing any number to 12367 (304E). This can be done either with function code 6 or 16.
2	Load the schedule into the setpoint scheduler block by writing the schedule number to registers 12321 and 12322 (3020 and 3021). This is a floating point register and requires a multiple register write (function code 16)
3	Read the header information desired -- registers 12331 (302A) - 12347 (303A) and register 12368 (304F) using function code 3.
4	Read the information for each segment desired using function code 3 -- registers 3200 - 322F for segment 1, 3230 - 325F for segment 2, etc.

This table contains Value Register addresses of SP Scheduler #1. ***See the Global Register Map - Table 6-1 for starting and ending addresses (hex) for SP Scheduler #2***

Table 6-18 Scheduler #1 Value Register Map Addresses

Address (hex)	Register (decimal)	Channel Number	Access	Notes
3000	12289	Scheduler Output 1	R	Floating Point in Engineering Units.
3002	12291	Scheduler Output 2	R	Floating Point in Engineering Units.
3004	12293	Scheduler Output 3	R	Floating Point in Engineering Units.
3006	12295	Scheduler Output 4	R	Floating Point in Engineering Units.
3008	12297	Scheduler Output 5	R	Floating Point in Engineering Units.
300A	12299	Scheduler Output 6	R	Floating Point in Engineering Units.
300C	12301	Scheduler Output 7	R	Floating Point in Engineering Units.
300E	12303	Scheduler Output 8	R	Floating Point in Engineering Units.
3010	12305	Scheduler Auxiliary Output 1	R	Floating Point in Engineering Units.
3012	12307	Scheduler Auxiliary Output 2	R	Floating Point in Engineering Units.
3014	12309	Scheduler Auxiliary Output 3	R	Floating Point in Engineering Units.
3016	12311	Scheduler Auxiliary Output 4	R	Floating Point in Engineering Units.
3018	12313	Scheduler Auxiliary Output 5	R	Floating Point in Engineering Units.
301A	12315	Scheduler Auxiliary Output 6	R	Floating Point in Engineering Units.
301C	12317	Scheduler Auxiliary Output 7	R	Floating Point in Engineering Units.
301E	12319	Scheduler Auxiliary Output 8	R	Floating Point in Engineering Units.

Address (hex)	Register (decimal)	Channel Number	Access	Notes
3020	12321	Current Program Number	R/W	Floating Point Indicates the present schedule number in use. This also allows entry of a schedule number from the HC900 stored schedule memory (if schedules have been stored in the controller) and will retrieve the schedule data for a display showing a SP schedule segment table. Typically, when recipes are downloaded from third party software, this will be the number generated by the Schedule Save Request parameter A write to this register loads the program into the scheduler function block; if 0 is written, the scheduler's schedule is cleared. Writing to this register is only permissible in reset or ready mode.
3022	12323	Current Segment Number	R/W	Floating Point; 1 thru Max Segment # A write changes the segment number.
3024	12325	Program Elapsed Time	R	Floating Point in Time Units Includes or runs when in Hold
3026	12327	Segment Time Remaining	R	Floating Point in Time Units
3028	12329	Schedule Save Request	R/W	Floating point. Assigns profile parameters downloaded to a schedule number such as 1. **This is required by the HC900 to be the last parameter downloaded.** This overwrites what is in this HC900 SP Schedule memory location on each download of a new SP Schedule. Saves the schedule into the archive. Writing to this register is prohibited in the run mode.
302A	12331	Guaranteed Soak Limit 1	R/W	Floating Point
302C	12333	Guaranteed Soak Limit 2	R/W	Floating Point
302E	12335	Guaranteed Soak Limit 3	R/W	Floating Point
3030	12337	Guaranteed Soak Limit 4	R/W	Floating Point
3032	12339	Guaranteed Soak Limit 5	R/W	Floating Point
3034	12341	Guaranteed Soak Limit 6	R/W	Floating Point
3036	12343	Guaranteed Soak Limit 7	R/W	Floating Point
3038	12345	Guaranteed Soak Limit 8	R/W	Floating Point
303A	12347	Jog Segment	R/W	Floating Point Defines segment for schedule to jog based on an enable to an input pin

Address (hex)	Register (decimal)	Channel Number	Access	Notes
3049	12362	Current Segment Events (Bit Packed)	R	Bit Packed Indicates status of events Bit 0: Event #1 : : Bit 15: Event #15 0: Event OFF 1: Event ON
304A	12363	Status (Bit Packed)	R	Bit Packed Bit 0: 1=Ready 1: 1=Run 2: 1=Hold 3: 1=End 4: 1=Time Units in Seconds 5: 1=Time Units in Minutes 6: 1=Time Units in Hours 7: If bit 2 Set 0: Operator hold 1: Guaranteed soak hold 8-15: Reserved
304B	12364	Start	W	Signed 16 bit integer Write to location Starts Schedule; Data ignored
304C	12365	Hold	W	Signed 16 bit integer Write to location Holds Schedule; Data ignored
304D	12366	Advance	W	Signed 16 bit integer Write to location Advances Schedule; Data ignored
304E	12367	Reset	W	Signed 16 bit integer Write to location Resets Schedule; Data ignored
304F	12368	Time Units	R/W	Bit Packed Bit 0: Reserved 2: hours 3-15: Unused

6.16 Scheduler Segment Register Map

A schedule can contain up to 50 segments. Each segment is made up of 48 (30 hex) registers. This table contains Segment Map addresses of SP Scheduler #1. ***See the Global Register Map - Table 6-1 for starting and ending addresses (hex) for SP Scheduler #2 Addresses.***

Table 6-19 Scheduler #1 Segment Register Map Addresses

Start Address (Hex)	Start Register (Decimal)	End Address (Hex)	End Register (Decimal)	Description
3200	12801	322F	12847	Scheduler #1 Segment 1
3230	12849	325F	12896	Scheduler #1 Segment 2
3260	12897	328F	12944	Scheduler #1 Segment 3
:		:	:	:
3B30	15153	3B5F	15200	Scheduler #1 Segment 50

6.17 Segment Register Map

The table below describes the registers that are part of a schedule segment. To determine the actual register address for a parameter within a segment, add the register offset to the start address of the segment.

Table 6-20 Segment Register Map Addresses

Register Offset within segment (Hex)	Register Offset within segment (Decimal)	Parameter Name	Access	Notes
0000	0000	Guaranteed Soak Type 1 (Bit Packed)	R/W	Bit Packed Bit 0: Off Bit 1: Low Bit 2: High Bit 3: Low & High Bit 4…15: Unused *Note 1*
0001	0001	Guaranteed Soak Type 2	R/W	*See Guaranteed Soak Type 1*
0002	0002	Guaranteed Soak Type 3	R/W	*See Guaranteed Soak Type 1*
0003	0003	Guaranteed Soak Type 4	R/W	*See Guaranteed Soak Type 1*
0004	0004	Guaranteed Soak Type 5	R/W	*See Guaranteed Soak Type 1*
0005	0005	Guaranteed Soak Type 6	R/W	*See Guaranteed Soak Type 1*
0006	0006	Guaranteed Soak Type 7	R/W	*See Guaranteed Soak Type 1*
0007	0007	Guaranteed Soak Type 8	R/W	*See Guaranteed Soak Type 1*
0008	0008	Events	R/W	Bit Packed Bit 0: Event #1 : : Bit 15: Event #15 0: Event OFF 1: Event ON *Note 2*
000A	0010	Time	R/W	Floating Point in seconds *Note 1*
000C	0012	Output #1 Ramp or Soak value	R/W	Floating Point *Note 1*
000E	0014	Output #2 Ramp or Soak value	R/W	Floating Point *Note 1*
0010	0016	Output #3 Ramp or Soak value	R/W	Floating Point *Note 1*
0012	0018	Output #4 Ramp or Soak value	R/W	Floating Point *Note 1*
0014	0020	Output #5 Ramp or Soak value	R/W	Floating Point *Note 1*
0016	0022	Output #6 Ramp or Soak value	R/W	Floating Point *Note 1*
0018	0024	Output #7 Ramp or Soak value	R/W	Floating Point *Note 1*

Register Offset within segment (Hex)	Register Offset within segment (Decimal)	Parameter Name	Access	Notes
001A	0026	Output #8 Ramp or Soak value	R/W	Floating Point *Note 1*
001C	0028	Soak value for Auxiliary Output #1	R/W	Floating Point *Note 1*
001E	0030	Soak value for Auxiliary Output #2	R/W	Floating Point *Note 1*
0020	0032	Soak value for Auxiliary Output #3	R/W	Floating Point *Note 1*
0022	0034	Soak value for Auxiliary Output #4	R/W	Floating Point *Note 1*
0024	0036	Soak value for Auxiliary Output #5	R/W	Floating Point *Note 1*
0026	0038	Soak value for Auxiliary Output #6	R/W	Floating Point *Note 1*
0028	0040	Soak value for Auxiliary Output #7	R/W	Floating Point *Note 1*
002A	0042	Soak value for Auxiliary Output #8	R/W	Floating Point *Note 1*
002C	0044	Recycle	R/W	Number of times to recycle Floating Point *Note 2*
002E	0046	Recycle Segment	R/W	Floating Point *Note 2*

Note 1: Writing to this register is not permissible in the run mode.
Note 2: Writing to this register is only permissible in reset or ready mode

Example for Determining a Segment Register

To change the ramp value for Output #6 in segment #5 of setpoint scheduler #1, the register address is determined as follows.

Step 1: Use Table 6-1 to determine the start address for scheduler #3's schedule. The value is 3200 Hex.

Step 2: Calculate the offset address for segment 5 in a schedule. This is calculated as:

Segment offset address = (segment number – 1) * 48

= (5-1) * 48

= 192 or C0 Hex

Step 3: Use the table above to determine the register offset for Output #6 ramp value. The value is 16 Hex.

Step 4: Calculate the address by adding the results of steps 1, 2, and 3 to determine the register address.

Register address = Schedule program #3's schedule start address
+ Segment 5 offset address
+ Output #6 ramp value register offset

= 3200 + C0 + 16

= 32D6

6.18 Sequencer Register Maps

Summary

The Sequencer Register Maps are listed according to category related to Sequencer operation.

Function Code Support

Read – Function Code 3

Write – Function Code 16 (10 hex)

Sequencer Parameters Register Map

Table 6-21 Sequence Parameters Register Maps

	Starting Address (Hex)	Starting Register (Decimal)	Ending Address (Hex)	Ending Address (Decimal)
Sequencer #1 Parameters	5AA0	23201	5AB5	23222
Sequencer #2 Parameters	5D00	23809	5D1F	23840
Sequencer #3 Parameters	5F60	24417	5F7F	24448
Sequencer #4 Parameters	A000	40961	A01F	40992

Sequencer Step 1 Table Register Map

Table 6-22 Sequence Step 1 Table Register Maps

	Starting Address (Hex)	Starting Register (Decimal)	Ending Address (Hex)	Ending Address (Decimal)
Sequencer #1 Step 1 Table	5AC0	23233	5CBF	23744
Sequencer #2 Step 1 Table	5D20	23841	5F1F	24532
Sequencer #3 Step 1 Table	5F80	24449	617f	24960
Sequencer #4 Step 1 Table	A020	40993	A21F	41504

Sequencer #1 State Table

Table 6-23 Sequence State Table Register Maps

	Starting Address (Hex)	Starting Register (Decimal)	Ending Address (Hex)	Ending Address (Decimal)
Sequencer #1 State Table	5CC0	23745	5CFF	23808
Sequencer #2 State Table	5F20	24353	5F5F	24416
Sequencer #3 State Table	6180	24961	61BF	25024
Sequencer #4 State Table	A220	41505	A25F	41568

6.19 Sequencer #1 Parameters Register Map

Summary

This table contains Parameter Map addresses of Sequencer #1. ***See the Sequencer Parameter Register Maps*** - Table 6-21 ***for starting and ending addresses (hex) for Sequencer #2 through Sequencer #4 Parameter Map Addresses.***

Table 6-24 Sequencer #1 Parameters Register Map Addresses

Address (hex)	Register (decimal)	Channel Number	Access	Notes
5AA0	23201	Not used	N/A	Not used
5AA1	23202	Save current sequence request	R/W	Integer, Saves current sequence to archive and assigns it to the number written (1-20).
5AA2	23203	Reset	W	Integer, Write to this location results in reset action and data is ignored.
5AA3	23204	Run	W	Integer, Write to this location starts run action and data is ignored.
5AA4	23205	Hold	W	Integer, Write to this location results in reset action and data is ignored.
5AA5	23206	Advance	W	Integer, Write to this location results in an advance to next step and data is ignored. *Writing to this register is only permissible in the hold mode.*
5AA6	23207	Not used	N/A	Not used
5AA7	23208	Step jumped to on Jog	R/W	Integer Step number for jog.
5AA8	23209	Current Sequence number	R	Integer Sequence number loaded.
5AA9	23210	Step Number	R	Integer value for current step number.
5AAA	23211	Load Sequence number	R/W	Integer, sequence number loaded to sequencer. *Writing to this register is only permissible in the Ready mode.*
5AAB	23212	Step number change request	R/W	Integer, go to a step number in any mode.
5AAC	23213	Outputs	R	Integer, Bit packed for 16 output values 0:OFF 1:ON
5AAD	23214	Time units	R/W	Integer, 0=hrs, 1=minutes.
5AAE	23215	Elapsed Sequence Time	R	Floating point value for elapsed sequence time in minutes.
5AB0	23217	Time Remaining in current step	R	Floating point value for time remaining in current step in minutes.

Address (hex)	Register (decimal)	Channel Number	Access	Notes
5AB2	23219	Sequence status	R	Integer value for bit packed current sequence state : Bit 0 = Ready Bit 1 = Run Bit 2 =Hold Bit 3 = Stop Bit 4 = Disable Bit 5 = time units (minutes) Bit 6 = time units (hours)
5AB4	23221	State	R	Integer value for current state number
5AB6	23223	Auxiliary Output	R	Floating point value auxiliary output

6.20 Sequencer #1 Step 1 Table Register Map

Summary

This table contains Step #1 Table Map addresses of Sequencer #1. ***See the Sequencer Step 1 Table Register Maps*** - Table 6-22 ***for starting and ending addresses (hex) for Sequencer #2 through Sequencer #4 Step #1 Table Map Addresses.***

Table 6-25 Sequencer #1 Step 1 Table Register Map Addresses

Address (hex)	Register (decimal)	Channel Number	Access	Notes
5AC0	23233	Time Next step	R/W	Integer value of the next step executed when time expires.
5AC1	23234	Event1 Next step	R/W	Integer value of the next step executed when configured event 1 transitions from OFF to ON.
5AC2	23235	Event2 Next step	R/W	Integer value of the next step executed when configured event 2 transitions from OFF to ON.
5AC3	23236	Advance Next step	R/W	Integer value of the next step executed when Advance transitions from OFF to ON.
5AC4	23237	Step Time	R/W	Floating value for time of present step.
5AC6	23239	Auxiliary Output	R/W	Floating Auxiliary analog value associated with the step.

Note: There are 64 instances of the step table (one for each step) for each Sequencer.

The addresses consumed by the full step table would be from 5AC0 thru 5CBFh.

6.21 Sequencer #1 State Table Register Map

Summary

This table contains State Table Map addresses of Sequencer #1. ***See the Sequencer State Table Register Maps*** - Table 6-23 ***for starting and ending addresses (hex) for Sequencer #2 through Sequencer #4 State Table Map Addresses.***

Table 6-26 Sequencer #1 State Table Register Map Addresses

Address (hex)	Register (decimal)	Channel Number	Access	Notes
5CC0	23745	State Number Step #1	R/W	Integer state number of the output definitions as configured between 1 and 50.
5CC1	23746	State Number Step #2	R/W	Integer state number of the output definitions as configured between 1 and 50.
5CC2	23747	State Number Step #3	R/W	Integer state number of the output definitions as configured between 1 and 50.

.
.
.

Address (hex)	Register (decimal)	Channel Number	Access	Notes
5CFF	23808	State Number Step #64	R/W	Integer state number of the output definitions as configured between 1 and 50.

6.22 Hand/OFF/Auto Control Group Register Map

Summary

This table contains Hand/Off/Auto Map addresses of HOA Group #1.

See the HOA Register Maps Table 6-1 ***for starting and ending addresses (hex) for Hand/Off/Auto Group #2 through Hand/Off/Auto Group #16 Map Addresses.***

The Modbus HOA number address for a HOA can also be obtained from the Hybrid Control Designer printout of "Block Modbus Addresses".

Function Code Support:

Reads – Function Code 3

Writes – Function Code 16 (10 hex) for preset of multiple registers (e.g., for floating point)

Writes – Function Code 6 for presetting an integer value

Table 6-27 HOA Control #1 Group Register Map

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
6600	26113	Status	R	Bit Packed Bits 0 – 3: Hand-Off-Auto State Bit 0: Off : 0=NO, 1=YES Bit 1: Hand: 0=NO, 1=YES Bit 2: Auto: 0=NO, 1=YES Bit 3: Bypass: 0=NO, 1=YES Bit 4: Request Output 0 = OFF, 1 = ON Bit 5: Local Source ON: 0=NO, 1=YES Bit 6: Remote Source ON: 0=NO, 1=YES Bit 7: Local and Remote ON: 0=NO, 1=YES Bit 7-15: Unused
6601	26114	Remote Off-state Change Request	W	Unsigned 16 Integer Data Ignored
6602	26115	Remote Hand-state Change Request	W	Unsigned 16 Integer Data Ignored
6603	26116	Remote Auto-state Change Request	W	Unsigned 16 Integer Data Ignored
6604	26117	Local Source	W	Unsigned 16 Integer Data Ignored
6605	26118	Remote Source	W	Unsigned 16 Integer Data Ignored
6606	26119	Local and Remote Source	W	Unsigned 16 Integer Data Ignored

6.23 Device Control Group Register Map

Summary

This table contains addresses for the Device Control group #1.

See the Device Control Register Maps in Table 6-1 for starting and ending addresses (hex) for Device Control Group #2 through Device Control Group #16 Map Addresses.

The Modbus Device Control number address for a Device Control can also be obtained from the Hybrid Control Designer printout of "Block Modbus Addresses".

Function Code Support:

Reads – Function Code 3

Writes – Function Code 16 (10 hex) for preset of multiple registers (e.g., for floating point)

Writes – Function Code 6 for presetting an integer value

Table 6-28 Device Control #1 Group Register Map

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
6B00	27393	Reset Request	W	Unsigned 16 Integer Data Ignored
6B01	27394	Status Indicator	R	Bit Packed Bits 0 – 6: Device Control State Bit 0: Ready: 0=NO, 1=YES Bit 1: Prestart: 0=NO, 1=YES Bit 2: Starting: 0=NO, 1=YES Bit 3: Running: 0=NO, 1=YES Bit 4: Stopping: 0=NO, 1=YES Bit 5: Disabled: 0=NO, 1=YES Bit 6: Failed: 0=NO, 1=YES Bit 7: Run Request Input State; 0=OFF, 1=ON Bit 8: Device Feedback Started; 0=NO, 1=YES Bit 9: Device Failed; 0=NO, 1=YES Bit 10: Automatic Reset; 0=Manual, 1=Auto Bit 11-15: Unused
6B02	27395	Remaining Delay Time	R	Floating Point in Seconds
6B04	27397	Start Delay	R/W	Floating Point in Seconds
6B06	27399	Stop Delay	R/W	Floating Point in Seconds
6B08	27401	Feedback Fail Delay	R/W	Floating Point in Seconds

6.24 Alternator Group Register Map

Summary

This section contains addresses for the Alternator #1group.

See the Alternator Register Maps in Table 6-1 for starting and ending addresses (hex) for Alternator #2 through Alternator #16 Map Addresses.

The Modbus Alternator number address for an Alternator can also be obtained from the Hybrid Control Designer printout of "Block Modbus Addresses".

Function Code Support:

Reads – Function Code 3

Writes – Function Code 16 (10 hex) for preset of multiple registers (e.g., for floating point)

Writes – Function Code 6 for presetting an integer value

ATTENTION

Output Order Sequence registers must be written in a single transaction. Duplicate sequence values (1-16) are not permitted.

Table 6-29 Alternator #1Group Register Map

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
6700	26369	Alternator Status	R	Bit Packed Bit 0: Enable: 0=NO, 1=YES Bit 1: Low Capacity 0=Meeting Capacity, 1=Low Capacity Bit 2-15: Unused
6701	26370	Device Ready (#1-16)	R	Bit Packed: Bit 0: Device #1 Ready: 0=NO, 1=YES Bit 1: Device #2 Ready: 0=NO, 1=YES Bit 2: Device #3 Ready: 0=NO, 1=YES Bit 3: Device #4 Ready: 0=NO, 1=YES Bit 4: Device #5 Ready: 0=NO, 1=YES Bit 5: Device #6 Ready: 0=NO, 1=YES Bit 6: Device #7 Ready: 0=NO, 1=YES Bit 7: Device #8 Ready: 0=NO, 1=YES Bit 8: Device #9 Ready: 0=NO, 1=YES Bit 9: Device #10 Ready: 0=NO, 1=YES Bit 10: Device #11 Ready: 0=NO, 1=YES Bit 11: Device #12 Ready: 0=NO, 1=YES Bit 12: Device #13 Ready: 0=NO, 1=YES Bit 13: Device #14 Ready: 0=NO, 1=YES Bit 14: Device #15 Ready: 0=NO, 1=YES Bit 15: device #16 Ready: 0=NO, 1=YES
6702	26371	Input Status #1-8	R	Bit Packed: Bit 0: Input #1 Enable: 0=NO, 1=YES

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
				Bit 1: Input #1 ON: 0=OFF, 1=ON Bit 2: Input #2 Enable: 0=NO, 1=YES Bit 3: Input #2 ON: 0=OFF, 1=ON Bit 4: Input #3 Enable: 0=NO, 1=YES Bit 5: Input #3 ON: 0=OFF, 1=ON Bit 6: Input #4 Enable: 0=NO, 1=YES Bit 7: Input #4 ON: 0=OFF, 1=ON Bit 8: Input #5 Enable: 0=NO, 1=YES Bit 9: Input #5 ON: 0=OFF, 1=ON Bit 10: Input #6 Enable: 0=NO, 1=YES Bit 11: Input #6 ON: 0=OFF, 1=ON Bit 12: Input #7 Enable: 0=NO, 1=YES Bit 13: Input #7 ON: 0=OFF, 1=ON Bit 14: Input #8 Enable: 0=NO, 1=YES Bit 15: Input #8 ON: 0=OFF, 1=ON
6703	26372	Input Status #9-16	R	Bit Packed: Bit 0: Input #9 Enable: 0=NO, 1=YES Bit 1: Input #9 ON: 0=OFF, 1=ON Bit 2: Input #10 Enable: 0=NO, 1=YES Bit 3: Input #10 ON: 0=OFF, 1=ON Bit 4: Input #11 Enable: 0=NO, 1=YES Bit 5: Input #11 ON: 0=OFF, 1=ON Bit 6: Input #12 Enable: 0=NO, 1=YES Bit 7: Input #12 ON: 0=OFF, 1=ON Bit 8: Input #13 Enable: 0=NO, 1=YES Bit 9: Input #13 ON: 0=OFF, 1=ON Bit 10: Input #14 Enable: 0=NO, 1=YES Bit 11: Input #14 ON: 0=OFF, 1=ON Bit 12: Input #15 Enable: 0=NO, 1=YES Bit 13: Input #15 ON: 0=OFF, 1=ON Bit 14: Input #16 Enable: 0=NO, 1=YES Bit 15: Input #16 ON: 0=OFF, 1=ON
6704	26373	Output Status #1-4	R	Bit Packed: Bit 0: Output #1 Enable: 0=NO, 1=YES Bit 1: Output #1 Used: 0=NO, 1=YES Bit 2: Output #1 ON: 0=OFF, 1=ON Bit 3: Unused Bit 4: Output #2 Enable: 0=NO, 1=YES Bit 5: Output #2 Used: 0=NO, 1=YES Bit 6: Output #2 ON: 0=OFF, 1=ON Bit 7: Unused Bit 8: Output #3 Enable: 0=NO, 1=YES

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
				Bit 9: Output #3 Used: 0=NO, 1=YES Bit 10: Output #3 ON: 0=OFF, 1=ON Bit 11: Unused Bit 12: Output #4 Enable: 0=NO, 1=YES Bit 13: Output #4 Used: 0=NO, 1=YES Bit 14: Output #4 ON: 0=OFF, 1=ON Bit 15: Unused
6705	26374	Output Status #5-8	R	Bit Packed: Bit 0: Output #5 Enable: 0=NO, 1=YES Bit 1: Output #5 Used: 0=NO, 1=YES Bit 2: Output #5 ON: 0=OFF, 1=ON Bit 3: Unused Bit 4: Output #6 Enable: 0=NO, 1=YES Bit 5: Output #6 Used: 0=NO, 1=YES Bit 6: Output #6 ON: 0=OFF, 1=ON Bit 7: Unused Bit 8: Output #7 Enable: 0=NO, 1=YES Bit 9: Output #7 Used: 0=NO, 1=YES Bit 10: Output # ON: 0=OFF, 1=ON Bit 11: Unused Bit 12: Output #8 Enable: 0=NO, 1=YES Bit 13: Output #8 Used: 0=NO, 1=YES Bit 14: Output # ON: 0=OFF, 1=ON Bit 15: Unused
6706	26375	Output Status #9-12	R	Bit Packed: Bit 0: Output #9 Enable: 0=NO, 1=YES Bit 1: Output #9 Used: 0=NO, 1=YES Bit 2: Output #9 ON: 0=OFF, 1=ON Bit 3: Unused Bit 4: Output #10 Enable: 0=NO, 1=YES Bit 5: Output #10 Used: 0=NO, 1=YES Bit 6: Output #10 ON: 0=OFF, 1=ON Bit 7: Unused Bit 8: Output #11 Enable: 0=NO, 1=YES Bit 9: Output # Used: 0=NO, 1=YES Bit 10: Output #11 ON: 0=OFF, 1=ON Bit 11: Unused Bit 12: Output #12 Enable: 0=NO, 1=YES Bit 13: Output #12 Used: 0=NO, 1=YES Bit 14: Output #12 ON: 0=OFF, 1=ON Bit 15: Unused
6707	26376	Output Status #13-	R	Bit Packed:

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
		16		Bit 0: Output #13 Enable: 0=NO, 1=YES Bit 1: Output #13 Used: 0=NO, 1=YES Bit 2: Output #13 ON: 0=OFF, 1=ON Bit 3: Unused Bit 4: Output # Enable: 0=NO, 1=YES Bit 5: Output #14 Used: 0=NO, 1=YES Bit 6: Output #14 ON: 0=OFF, 1=ON Bit 7: Unused Bit 8: Output #15 Enable: 0=NO, 1=YES Bit 9: Output #15 Used: 0=NO, 1=YES Bit 10: Output #15 ON: 0=OFF, 1=ON Bit 11: Unused Bit 12: Output #16 Enable: 0=NO, 1=YES Bit 13: Output #16 Used: 0=NO, 1=YES Bit 14: Output #16 ON: 0=OFF, 1=ON Bit 15: Unused
6708	26377	Request Style	*R*	Bit Packed: Bit 0: Direct: 0=NO, 1=YES Bit 1: Rotary: 0=NO, 1=YES Bit 2: FOFO: 0=NO, 1=YES Bit 3: Fixed: 0=NO, 1=YES Bit 4-15: Unused
6709	26378	Direct Request	W	Unsigned 16 Integer Data Ignored
670A	26379	Rotary Request	W	Unsigned 16 Integer Data Ignored
670B	26380	First on/First off Request	W	Unsigned 16 Integer Data Ignored
670C	26381	Fixed Request	W	Unsigned 16 Integer Data Ignored
670D	26382	Advance Request	W	This does not apply if current style is Direct or if Advance Active configuration parameter is set to OFF Unsigned 16 Integer Data Ignored
670E	26383	Input Count	R	Floating Point in Counts
6710	26385	Output On-Delay	R/W	Floating Point in Seconds
6712	26387	Output Off-Delay	R/W	Floating Point in Seconds
6714	26389	Input Enable (1-16)	R/W	Bit Packed: Bit 0: Device #1 Enable: 0=NO, 1=YES Bit 1: Device #2 Enable: 0=NO, 1=YES Bit 2: Device #3 Enable: 0=NO, 1=YES

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
				Bit 3: Device #4 Enable: 0=NO, 1=YES Bit 4: Device #5 Enable: 0=NO, 1=YES Bit 5: Device #6 Enable: 0=NO, 1=YES Bit 6: Device #7 Enable: 0=NO, 1=YES Bit 7: Device #8 Enable: 0=NO, 1=YES Bit 8: Device #9 Enable: 0=NO, 1=YES Bit 9: Device #10 Enable: 0=NO, 1=YES Bit 10: Device #11 Enable: 0=NO, 1=YES Bit 11: Device #12 Enable: 0=NO, 1=YES Bit 12: Device #13 Enable: 0=NO, 1=YES Bit 13: Device #14 Enable: 0=NO, 1=YES Bit 14: Device #15 Enable: 0=NO, 1=YES Bit 15: Device #16 Enable: 0=NO, 1=YES
6715	26390	Output Enable (1-16)	R/W	Bit Packed: Bit 0: Device #1 Enable: 0=NO, 1=YES Bit 1: Device #2 Enable: 0=NO, 1=YES Bit 2: Device #3 Enable: 0=NO, 1=YES Bit 3: Device #4 Enable: 0=NO, 1=YES Bit 4: Device #5 Enable: 0=NO, 1=YES Bit 5: Device #6 Enable: 0=NO, 1=YES Bit 6: Device #7 Enable: 0=NO, 1=YES Bit 7: Device #8 Enable: 0=NO, 1=YES Bit 8: Device #9 Enable: 0=NO, 1=YES Bit 9: Device #10 Enable: 0=NO, 1=YES Bit 10: Device #11 Enable: 0=NO, 1=YES Bit 11: Device #12 Enable: 0=NO, 1=YES Bit 12: Device #13 Enable: 0=NO, 1=YES Bit 13: Device #14 Enable: 0=NO, 1=YES Bit 14: Device #15 Enable: 0=NO, 1=YES Bit 15: Device #16 Enable: 0=NO, 1=YES
6716	26391	Output Order Sequence #1	R	Unsigned 16 integer
6717	26392	Output Order Sequence #2	R	Unsigned 16 integer
6718	26393	Output Order Sequence #3	R	Unsigned 16 integer
6719	26394	Output Order Sequence #4	R	Unsigned 16 integer
671A	26395	Output Order Sequence #5	R	Unsigned 16 integer

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
671B	26396	Output Order Sequence #6	R	Unsigned 16 integer
671C	26397	Output Order Sequence #7	R	Unsigned 16 integer
671D	26398	Output Order Sequence #8	R	Unsigned 16 integer
671E	26399	Output Order Sequence #9	R	Unsigned 16 integer
671F	26400	Output Order Sequence #10	R	Unsigned 16 integer
6720	26401	Output Order Sequence #11	R	Unsigned 16 integer
6721	26402	Output Order Sequence #12	R	Unsigned 16 integer
6722	26403	Output Order Sequence #13	R	Unsigned 16 integer
6723	26404	Output Order Sequence #14	R	Unsigned 16 integer
6724	26405	Output Order Sequence #15	R	Unsigned 16 integer
6725	26406	Output Order Sequence #16	R	Unsigned 16 integer

6.25 Output Order Sequence Scratch Pad Register Map

Summary

Output Order Sequence Scratch Pad registers in the Alternator Parameters are modified by using the output order sequence scratchpad and the sequence write request.

This section contains addresses for the Output order Sequence Scrarch pad.

Function Code Support:

Reads – Function Code 3

Writes – Function Code 16 (10 hex) for preset of multiple registers (e.g., for floating point)

Writes – Function Code 6 for presetting an integer value

Table 6-30 Output Order Sequence Scratch Pad Group Register Map

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
6A00	27137	Output Order Sequence #1	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A01	27138	Output Order Sequence #2	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A02	27139	Output Order Sequence #3	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A03	27140	Output Order Sequence #4	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A04	27141	Output Order Sequence #5	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A05	27142	Output Order Sequence #6	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A06	27143	Output Order Sequence #7	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A07	27144	Output Order Sequence #8	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A08	27145	Output Order Sequence #9	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A09	27146	Output Order Sequence #10	R/W	Unsigned 16 integer Only for Fixed and Direct styles.

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
				(See attention below)
6A0A	27147	Output Order Sequence #11	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A0B	27148	Output Order Sequence #12	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A0C	27149	Output Order Sequence #13	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A0D	27150	Output Order Sequence #14	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A0E	27151	Output Order Sequence #15	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A0F	27152	Output Order Sequence #16	R/W	Unsigned 16 integer Only for Fixed and Direct styles. (See attention below)
6A10	27153	Sequence Write Req. (moves the scratch pad into an alternator block)	W	Unsigned 16 integer (0-16) 0=Clears the scratch pad 1-16 = Alternator block's Modbus reference number

ATTENTION

Output Order Sequence registers are modified by using the output order sequence scratchpad and the sequence write request.

6.26 Stage Group Register Map

Summary

This section contains addresses for the Stage #1 Group.

See the Stage Group Register Maps in Table 6-1 for starting and ending addresses (hex) for Stage Group #2 through Stage Group #8 Map Addresses.

The Modbus Stage number address for a Stage Group can also be obtained from the Hybrid Control Designer printout of "Block Modbus Addresses".

Function Code Support:

Reads – Function Code 3

Writes – Function Code 16 (10 hex) for preset of multiple registers (e.g., for floating point)

Writes – Function Code 6 for presetting an integer value

Table 6-31 Stage Group #1 Register Map

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
6200	25089	PV1	R	Floating Point in Engineering Units
6202	25091	PV2	R	Floating Point in Engineering Units
6204	25093	Stage #1-4 Override Status	R	Bit Packed: Bit 0: Stage #1 Override Active: 0=NO, 1=YES Bit 1: Stage #1 Override ON: 0=OFF, 1=ON Bit 2: Stage #2 Override Active: 0=NO, 1=YES Bit 3: Stage #2 Override ON: 0=OFF, 1=ON Bit 4: Stage #3 Override Active: 0=NO, 1=YES Bit 5: Stage #3 Override ON: 0=OFF, 1=ON Bit 6: Stage #4 Override Active: 0=NO, 1=YES Bit 7: Stage #4 Override ON: 0=OFF, 1=ON Bits 8-15: Unused
6205	25094	Stage #1-4 PV On/Off Comparison	R	Bit Packed: Bit 0: Stage #1 On Comparitor: 0=PV1, 1=PV2 Bit 1: Stage #1 Off Comparitor: 0=PV1, 1=PV2 Bit 2: Stage #2 On Comparitor: 0=PV1, 1=PV2 Bit 3: Stage #2 Off Comparitor: 0=PV1, 1=PV2 Bit 4: Stage #3 On Comparitor: 0=PV1, 1=PV2 Bit 5: Stage #3 Off Comparitor: 0=PV1, 1=PV2 Bit 6: Stage #4 On Comparitor: 0=PV1, 1=PV2 Bit 7: Stage #4 Off Comparitor: 0=PV1, 1=PV2 Bit 8-15: Unused
6206	25095	Stage #1-4 Previous/Next Interlock	R	Bit Packed: Bit 0: Stage #1 Interlock with Previous Stage 0=NO, 1=YES Bit 1: Stage #1 Interlock with Next Stage 0=NO, 1=YES

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
				Bit 2: Stage #1 Error with Interlocks 0=NO, 1=YES Bit 3: Stage #2 Interlock with Previous Stage 0=NO, 1=YES Bit 4: Stage #2 Interlock with Next Stage 0=NO, 1=YES Bit 5: Stage #2 Error with Interlocks 0=NO, 1=YES Bit 6: Stage #3 Interlock with Previous Stage 0=NO, 1=YES Bit 7: Stage #3 Interlock with Next Stage 0=NO, 1=YES Bit 8: Stage #3 Error with Interlocks 0=NO, 1=YES Bit 9: Stage #4 Interlock with Previous Stage 0=NO, 1=YES Bit 10: Stage #4 Interlock with Next Stage 0=NO, 1=YES Bit 11: Stage #4 Error with Interlocks 0=NO, 1=YES Bit 12 - 15: Unused
6207	25096	Stage #1-4 Output Status (Request)	*R*	Bit Packed: Bit 0: Stage #1 Output Enable: 0=NO, 1=YES Bit 1: Stage #1 Output ON: 0=OFF, 1=ON Bit 2: Stage #2 Output Enable: 0=NO, 1=YES Bit 3: Stage #2 Output ON: 0=OFF, 1=ON Bit 4: Stage #3 Output Enable: 0=NO, 1=YES Bit 5: Stage #3 Output ON: 0=OFF, 1=ON Bit 6: Stage #4 Output Enable: 0=NO, 1=YES Bit 7: Stage #4 Output ON: 0=OFF, 1=ON Bit 8-15: Unused
6208	25097	Stage #1 ON Setpoint	R/W	Floating Point in Engineering Units
620A	25099	Stage #2 ON Setpoint	R/W	Floating Point in Engineering Units
620C	25101	Stage #3 ON Setpoint	R/W	Floating Point in Engineering Units
620E	25103	Stage #4 ON Setpoint	R/W	Floating Point in Engineering Units
6210	25105	Stage #1 OFF Setpoint	R/W	Floating Point in Engineering Units
6212	25107	Stage #2 OFF Setpoint	R/W	Floating Point in Engineering Units

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
6214	25109	Stage #3 OFF Setpoint	R/W	Floating Point in Engineering Units
6216	25111	Stage #4 OFF Setpoint	R/W	Floating Point in Engineering Units
6218	25113	Stage #1 Latch Delay	R/W	Floating Point in Seconds
621A	25115	Stage #2 Latch Delay	R/W	Floating Point in Seconds
621C	25117	Stage #3 Latch Delay	R/W	Floating Point in Seconds
621E	25119	Stage #4 Latch Delay	R/W	Floating Point in Seconds
6220	25121	Stage #1 Unlatch Delay	R/W	Floating Point in Seconds
6222	25123	Stage #2 Unlatch Delay	R/W	Floating Point in Seconds
6224	25125	Stage #3 Unlatch Delay	R/W	Floating Point in Seconds
6226	25127	Stage #4 Unlatch Delay	R/W	Floating Point in Seconds

6.27 Ramp Group Register Map

Summary

This section contains addresses for the Ramp #1 Group.

See the Ramp Group Register Maps in Table 6-1 for starting and ending addresses (hex) for Ramp Group #2 through Ramp Group #8 Map Addresses.

The Modbus Ramp number address for a Ramp Group can also be obtained from the Hybrid Control Designer printout of "Block Modbus Addresses".

Function Code Support:

Reads – Function Code 3

Writes – Function Code 16 (10 hex) for preset of multiple registers (e.g., for floating point)

Writes – Function Code 6 for presetting an integer value

Table 6-32 Ramp Group #1 Register Map

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
6400	25601	PV	R	Floating Point in Engineering Units
6402	25603	Ramp Group Output	R	Floating Point in Engineering Units
6404	25605	Ramp Group Default	R	Floating Point in Engineering Units
6406	25607	Ramp Lag Time	R/W	Floating Point in Seconds
6408	25609	Ramp Transfer Up	R/W	Floating Point in Engineering Unit/Second
640A	25611	Ramp Transfer Dn	R/W	Floating Point in Engineering Unit/Second
640C	25613	Ramp #1-4 Enable	R	Bit Packed: Bit 0: Ramp #1 Enable: 0=NO, 1=YES Bit 1: Ramp #2 Enable: 0=NO, 1=YES Bit 2: Ramp #3 Enable: 0=NO, 1=YES Bit 3: Ramp #4 Enable: 0=NO, 1=YES Bit 4-15: Unused
640D	25614	Ramp #1-4 Override Status	R	Bit Packed Bit 0: Ramp #1 Override Active: 0= NO, 1=YES Bit 1: Ramp #1 Override to High Limit: 0=Low Limit, 1= High Limit Bit 2: Ramp #2 Override Active: 0= NO, 1=YES Bit 3: Ramp #2 Override to High Limit: 0=Low Limit, 1= High Limit Bit 4: Ramp #3 Override Active: 0= NO, 1=YES Bit 5: Ramp #3 Override to High Limit: 0=Low Limit, 1= High Limit Bit 6: Ramp #4 Override Active: 0= NO, 1=YES Bit 7: Ramp #4 Override to High Limit:

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
				0=Low Limit, 1= High Limit Bit 8-15: Unused
640E	25615	Ramp #1 Output Scale High	R/W	Floating Point in Engineering Units
6410	25617	Ramp #2 Output Scale High	R/W	Floating Point in Engineering Units
6412	25619	Ramp #3 Output Scale High	R/W	Floating Point in Engineering Units
6414	25621	Ramp #4 Output Scale High	R/W	Floating Point in Engineering Units
6416	25623	Ramp #1 Output Scale Low	R/W	Floating Point in Engineering Units
6418	25625	Ramp #2 Output Scale Low	R/W	Floating Point in Engineering Units
641A	25627	Ramp #3 Output Scale Low	R/W	Floating Point in Engineering Units
641C	25629	Ramp #4 Output Scale Low	R/W	Floating Point in Engineering Units
641E	25631	Ramp #1 Input Limit High	R/W	Floating Point in Engineering Units
6420	25633	Ramp #2 Input Limit High	R/W	Floating Point in Engineering Units
6422	25635	Ramp #3 Input Limit High	R/W	Floating Point in Engineering Units
6424	25637	Ramp #4 Input Limit High	R/W	Floating Point in Engineering Units
6426	25639	Ramp #1 Input Limit Low	R/W	Floating Point in Engineering Units
6428	25641	Ramp #2Input Limit Low	R/W	Floating Point in Engineering Units
642A	25643	Ramp #3Input Limit Low	R/W	Floating Point in Engineering Units
642C	25645	Ramp #4Input Limit Low	R/W	Floating Point in Engineering Units

6.28 User Defined Registers

Each register can be assigned to a signal tag or variable defined in a function block diagram. The data type is configurable to be floating-point, signed-16, unsigned-16, signed-32, or unsigned-32. Floating point, signed-32, and unsigned-32 must start on an even hex address. 32-bit data uses the same byte-order format as defined for the port's double-register format. Writing a non-zero to a digital variable will set the variable to ON regardless of the data type. Single register writes to 32-bit digital variables are permitted. Integer data is scaled based on the number of decimal places defined for the variable/signal.

Table 6-33 User Defined Register Map

Address (hex)	Register (decimal)	Parameter Name	Access	Notes
B000 to B3E7	85057 to 86056	User defined signal or variable	Signal: R Variable: R/W	Read only if a signal tag Read/write if a variable

Index

Sales and Service

For application assistance, current specifications, pricing, or name of the nearest Authorized Distributor,contact one of the offices below.

ARGENTINA
Honeywell S.A.I.C.
Belgrano 1156
Buenos Aires
Argentina
Tel. : 54 1 383 9290

ASIA PACIFIC
Honeywell Asia
Pacific Inc.
Room 3213-3225
Sun Kung Kai Centre
N° 30 Harbour Road
Wanchai
Hong Kong
Tel. : 852 829 82 98

AUSTRALIA
Honeywell Limited
5 Thomas Holt Drive
North Ryde Sydney
Nsw Australia 2113
Tel. : 61 2 353 7000

AUSTRIA
Honeywell Austria
G.M.B.H.
Handelskai 388
A1020 Vienna
Austria
Tel. : 43 1 727 800

BELGIUM
Honeywell S.A.
3 Avenue De Bourget
B-1140 Brussels
Belgium
Tel. : 32 2 728 27 11

BRAZIL
HONEYWELL DO
Brazil
And Cia
Rua Jose Alves Da
Chunha
Lima 172
Butanta
05360.050 Sao Paulo
Sp
Brazil
Tel. : 55 11 819 3755

BULGARIA
HONEYWELL EOOD
14, Iskarsko Chausse
POB 79
BG- 1592 Sofia
BULGARIA
Tel : 359-791512/
794027/ 792198

CANADA
Honeywell Limited
The Honeywell Centre
300 Yorkland Blvd.
Toronto, Ontario
M2j 1s1
Canada
Tel.: 800 461 0013
Fax:: 416 502 5001

CZECH REPUBLIC
HONEYWELL, Spol.S.R.O.
Budejovicka 1
140 21 Prague 4
Czech Republic
Tel. : 42 2 6112 3434

DENMARK
HONEYWELL A/S
Automatikvej 1
DK 2860 Soeborg
DENMARK
Tel. : 45 39 55 56 58

FINLAND
HONEYWELL OY
Ruukintie 8
FIN-02320 ESPOO 32
FINLAND
Tel. : 358 0 3480101

FRANCE
HONEYWELL S.A.
Bâtiment « le Mercury »
Parc Technologique de St
Aubin
Route de l'Orme
(CD 128)
91190 SAINT-AUBIN
FRANCE
Tel. from France:
01 60 19 80 00
From other countries:
33 1 60 19 80 00

GERMANY
HONEYWELL AG
Kaiserleistrasse 39
D-63067 OFFENBACH
GERMANY
Tel. : 49 69 80 64444

HUNGARY
HONEYWELL Kft
Gogol u 13
H-1133 BUDAPEST
HUNGARY
Tel. : 36 1 451 43 00

ICELAND
HONEYWELL
Hataekni .hf
Armuli 26
PO Box 8336
128 reykjavik
Iceland
Tel : 354 588 5000

ITALY
HONEYWELL S.p.A.
Via P. Gobetti, 2/b
20063 Cernusco Sul
Naviglio
ITALY
Tel. : 39 02 92146 1

MEXICO
HONEYWELL S.A. DE
CV
AV. CONSTITUYENTES
900
COL. LOMAS ALTAS
11950 MEXICO CITY
MEXICO
Tel : 52 5 259 1966

THE NETHERLANDS
HONEYWELL BV
Laaderhoogtweg 18
1101 EA AMSTERDAM
ZO
THE NETHERLANDS
Tel : 31 20 56 56 911

NORWAY
HONEYWELL A/S
Askerveien 61
PO Box 263
N-1371 ASKER
NORWAY
Tel. : 47 66 76 20 00

POLAND
HONEYWELL Sp.z.o.o
Ul Domaniewksa 41
02-672 WARSAW
POLAND
Tel. : 48 22 606 09 00

PORTUGAL
HONEYWELL
PORTUGAL LDA
Edificio Suecia II
Av. do Forte nr 3 - Piso 3
2795 CARNAXIDE
PORTUGAL
Tel. : 351 1 424 50 00

REPUBLIC OF IRELAND
HONEYWELL
Unit 1
Robinhood Business
Park
Robinhood Road
DUBLIN 22
Republic of Ireland
Tel. : 353 1 4565944

REPUBLIC OF SINGAPORE
HONEYWELL PTE LTD
BLOCK 750E CHAI
CHEE ROAD
06-01 CHAI CHEE IND.PARK
1646 SINGAPORE
REP. OF SINGAPORE
Tel. : 65 2490 100

REPUBLIC OF SOUTH AFRICA
HONEYWELL
Southern Africa
PO BOX 138
Milnerton 7435
REPUBLIC OF SOUTH
AFRICA
Tel. : 27 11 805 12 01

ROMANIA
HONEYWELL Office
Bucharest
147 Aurel Vlaicu Str.,
Sc.Z.,
Apt 61/62
R-72921 Bucharest
ROMANIA
Tel : 40-1 211 00 76/
211 79

RUSSIA
HONEYWELL INC
4 th Floor Administrative
Builiding of AO "Luzhniki"
Management
24 Luzhniki
119048 Moscow
RUSSIA
Tel : 7 095 796 98 00/01

SLOVAKIA
HONEYWELL Ltd
Mlynske nivy 73
PO Box 75
820 07 BRATISLAVA 27
SLOVAKIA
Tel. : 421 7 52 47 400/425

SPAIN
HONEYWELL S.A
Factory
Josefa Valcarcel, 24
28027 MADRID
SPAIN
Tel. : 34 91 31 3 61 00

SWEDEN
HONEYWELL A.B.
S-127 86 Skarholmen
STOCKHOLM
SWEDEN
Tel. : 46 8 775 55 00

SWITZERLAND
HONEYWELL A.G.
Hertistrasse 2
8304 WALLISELLEN
SWITZERLAND
Tel. : 41 1 831 02 71

TURKEY
HONEYWELL A.S.
Caryiryolu Sok No. 7
Ucgen Plaza, Kat 5-6-7
Icerenkoy 81120 Instanbul
Turkey
Tel (90-216) 575 66 00

UNITED KINGDOM
HONEYWELL
Honeywell House
Arlington Business Park
Bracknell,
Berkshire
RG12 1EB
Tel: +44 (0) 1344 656000

U.S.A.
HONEYWELL INC.
INDUSTRIAL PROCESS
CONTROLS
1100 VIRGINIA DRIVE
PA 19034-3260
FT. WASHINGTON
U.S.A.
Tel. : 1-800-343-0228

VENEZUELA
HONEYWELL CA
APARTADO 61314
1060 CARACAS
VENEZUELA
Tel. : 58 2 239 0211

Honeywell

Industrial Measurement and Control
Honeywell
1100 Virginia Drive
Fort Washington, PA 19034

51-52-25-111 Rev. 6 0106 Printed in USA www.honeywell.com/imc