

# **TECHNICAL MANUAL**

## **Modbus User's Guide**

DHC Series Controllers with OCM-102

 **PEAKTRONICS**



## CONTENTS

	page
<b>SECTION I – Overview</b>	1
System Overview	
DHC Control Panel	
Control Hierarchy	
OCM-102 Operation	
Modbus Functions \$03 and \$06	
Modbus Function \$08	
PACS <sup>®</sup> Funcion \$41	
PACS <sup>®</sup> Parameters	
<b>SECTION II - Modbus Registers</b>	5
OCM-102 PACS <sup>®</sup> Off Line Timer Value	
Aux Closed Position	
Aux Open Position	
PACS <sup>®</sup> Command Input	
PACS <sup>®</sup> Control Register	
PACS <sup>®</sup> Timeout Setting	
Analog Command Input Position	
Average % Duty Cycle	
Maximum Deviation	
Hours of Operation	
Cycles of Operation	
Configuration / Alarm Flags	
Override Control / Mode	
Model Code / Serial Number	
OCM-102 Version	
Battery Voltage	
Average Motor Current	
Motor Current Trip Setting	
Alarm Flags / Alarm Flags 2	
<b>SECTION III - PACS<sup>®</sup> Read Only Parameters</b>	15
Analog Command Input	
Battery Voltage	
Average Motor Current	
Motor Current Trip Setting	
Position	
Alarm Flags 2	
Average % Duty Cycle	
Maximum Deviation	

Hours of Operation  
Cycles of Operation  
Alarm Flags  
Mode  
Configuration  
Override Control  
Serial Number  
Serial No. MSB / Model Code

<b>SECTION IV - PACS<sup>®</sup> Read/Write Parameters</b>	<b>23</b>
Aux Closed Position	
Aux Open Position	
PACS <sup>®</sup> Command Input	
PACS <sup>®</sup> Control Register	
PACS <sup>®</sup> Timeout Setting	
PACS <sup>®</sup> Reset	
User Non-volatile Memory	
Unassigned Locations	
<b>SECTION V - APPLICATION INFORMATION</b>	<b>28</b>
Bus Timing	
Using a Digital Command Type	
Overriding an Analog Command Type	
<b>APPENDIX A - PACS<sup>®</sup> Reference Guide</b>	<b>32</b>
<b>APPENDIX B - ASCII Conversion Table</b>	<b>33</b>

# SECTION I

## Overview

The scope of this document is to provide a description of digital parameters available in a Peaktronics DHC Series Controller, and how to access those parameters via a Modbus connection. DHC Series Controllers covered by this manual are listed below. New models may get added to this list; consult the latest revision of this manual for the most current list.

<b>MODEL No.</b>	<b>DESCRIPTION</b>
DHC-100	Digital High-Resolution Controller, 5A 117VAC
DHC-100A	Digital High-Resolution Controller, 5A 234VAC
DHC-100B	Digital High-Resolution Controller, 5A 24VAC
DHC-100C	CE Ready Digital High Resolution Controller, 5A 117VAC
DHC-100D	CE Digital High Resolution Controller, 5A 234VAC
DHC-100E	CE Ready Digital High Resolution Controller, 5A 24VAC
DHC-200	Digital High-Resolution Solenoid Controller, 117VAC
DHC-200A	Digital High-Resolution Solenoid Controller, 234VAC
DHC-200B	Digital High-Resolution Solenoid Controller, 24VAC
DHC-200C	EMI Hardened Digital High Resolution Solenoid Controller, 117VAC
DHC-200D	EMI Hardened Digital High Resolution Solenoid Controller, 234VAC
DHC-200E	EMI Hardened Digital High Resolution Solenoid Controller, 24VAC
DHC-300	Digital High-Resolution 3-Phase Controller, 117VAC
DHC-300A	Digital High-Resolution 3-Phase Controller, 234VAC
DHC-300B	Digital High-Resolution 3-Phase Controller, 24VAC
DHC-400	DC Digital High-Resolution Controller, 10-30VDC

## SYSTEM OVERVIEW

The **DHC Controller** is a high performance digital positioner designed to control an actuator that is usually used in automated valve applications. The DHC has an embedded PACS<sup>®</sup> Level 1 Slave port that provides digital communications through its option module connector.

To connect to Modbus, the DHC requires an **OCM-102 Modbus Option Module**. The OCM-102 acts as an interface between Modbus and PACS<sup>®</sup> (Peaktronics Asynchronous Communications System). The OCM-102 merely plugs into the DHC controller and provides the necessary connections for the bus. It also provides the means for configuring the module for various bus settings such as RTU/ASCII mode, Parity, baud rate, and node address.

The data sheets for the DHC Controller and the OCM-102 Modbus Option Module provide complete details of the use, function, and setup of these devices. Familiarity with each device may facilitate a better understanding of the digital parameters discussed in this manual. While this manual provides detailed examples of PACS<sup>®</sup> commands that can be used to access or control the DHC, a more comprehensive description of all PACS<sup>®</sup> commands can be found in "The PACS<sup>®</sup> Standard" technical manual, available from Peaktronics.

will not act on those parameters until the DHC is returned to the AUTO mode. Section II discusses these parameters in detail.

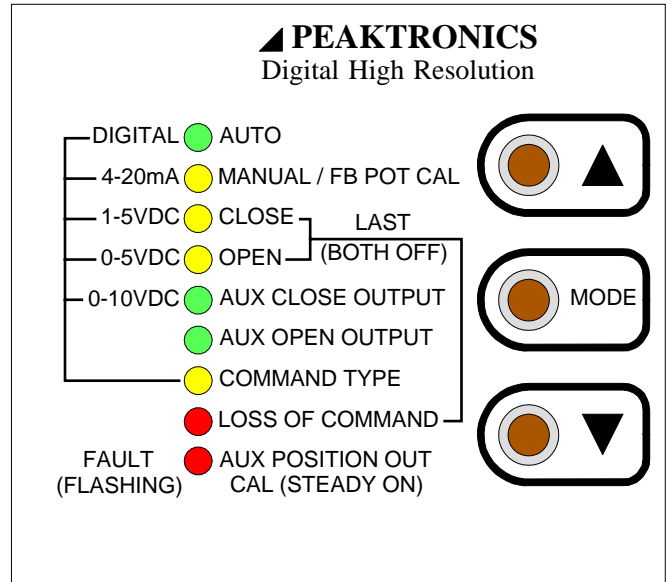


Figure 1 - DHC-100/200/300 Control Panel

## DHC CONTROL PANEL

Many of the DHC parameters relate to functions that are set from the DHC controller's front panel. These functions are referenced in the various discussions of the digital parameters. Figures 1 and 2 provide a visual aid of the DHC control panel.

## CONTROL HIERARCHY

All DHC parameters can be monitored, or read, any time the DHC is powered and on line. Reading parameters does not interfere with the DHC's operation or settings. Some parameters can be changed, or written, via the bus; these are referred to as *Read/Write Parameters*. The DHC will prohibit writing of various parameters to protect against accidental interference with critical DHC functions or settings; these are referred to as *Read Only Parameters*.

The DHC controller has a number of modes of operation. With the exception of the AUTO mode, all the modes are intended for either setup or trouble shooting tasks. In those cases, it usually means that a human operator has decided to control the actuator at the actuator site. To prevent confusion, or even a hazard, to the human operator, the DHC will ignore read/write parameters from the bus that run the actuator. The automation system can continue reading or writing such parameters, but the DHC

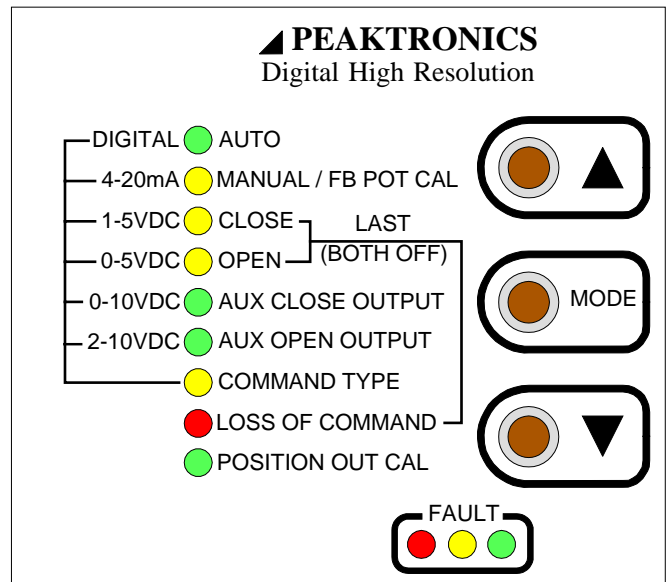


Figure 2 - DHC-400 Control Panel

## OCM-102 OPERATION

The OCM-102 is a Modbus slave device addressable from 1 to 247 and supports the following Modbus Functions:

\$03	Read Holding Registers
\$06	Write Single Register
\$08	Diagnostics
\$41	PACS® (user defined)

The DHC controller is a PACS® Slave device that is directly accessible using PACS® commands (see Appendix A) with the user defined Function \$41 (65 decimal), and is referred to here as the PACS® Function. Upon receiving a Function \$41 command, the OCM-102 switches operation to a gateway that allows the user to directly communicate to the DHC controller.

Upon receiving a Function \$03 or \$06 command, the OCM-102 switches operation to a Modbus interface. In this mode of operation, the OCM-102 maintains a virtual image of the DHC controller's parameters that allows fast access to these parameters using conventional Modbus commands. On power up, the OCM-102 defaults to the Modbus interface mode of operation.

Function \$08 is strictly diagnostic and actually communicates to the OCM-102 rather than the DHC controller. Function \$08 does not switch the mode of operation previously set by Function \$03, \$06, or \$41.

When switching to a gateway, by using Function \$41, the OCM-102 disables updates to the virtual image used for Functions \$03 and \$06. This is done to prevent interference with the PACS® commands used with Function \$41. If Function \$03 is used to switch back to a Modbus interface, an error response with exception code \$0B will be returned because the OCM-102 will need to initialize the virtual image before it can return good data. The initialization process typically requires 75msec, so the exception response will continue to be given until the initialization is completed.

## MODBUS FUNCTIONS \$03 and \$06

The DHC Series parameters are summarized in Table 1. Section II provides a complete description of the *data bytes* associated with each parameter.

Register Addresses 0000 to 0017<sub>hex</sub> are readable parameters and can be read using Function \$03. An error response with exception code \$03 is returned if the **Starting Register Address** is greater than 0017<sub>hex</sub>, or if the **Number of Registers** requested results in a **Register Address** that is greater than 0017<sub>hex</sub>.

Register Addresses 0000 to 0005 are writable parameters that can be written using Function \$06. An

error response with exception code \$03 is returned if Function \$06 requests an address higher than 0005.

## MODBUS FUNCTION \$08

The OCM-102 supports the Modbus Diagnostics Function (\$08) and can be placed in the Listen Only Mode. While in the Listen Only Mode, the OCM-102 will not pass bus commands to the DHC. If the OCM-102 receives a Broadcast command (Modbus address \$00), it will pass all bus commands it receives to the DHC, provided that the OCM-102 is not in the Listen Only Mode.

**Table 1 - Modbus Functions \$03 and \$06 Registers**

REGISTER NUMBER	REGISTER ADDRESS	PARAMETER
1	0000	OCM-102 Off Line Timeout Setting
2	0001	Aux Closed Position
3	0002	Aux Open Position
4	0003	PACS® Command Input
5	0004	PACS® Control Register
6	0005	PACS® Timeout Setting
7	0006	Analog Command Input
8	0007	Position
9	0008	Average % Duty Cycle
10	0009	Maximum Deviation
11	000A	Hours of Operation (MSW)
12	000B	Hours of Operation (LSW)
13	000C	Cycles of Operation (MSW)
14	000D	Cycles of Operation (LSW)
15	000E	Configuration / Alarm Flags
16	000F	Override Control / Mode
17	0010	Model Code / Serial Number (MSW)
18	0011	Model Code / Serial Number (LSW)
19	0012	OCM-102 Version (MSW)
20	0013	OCM-102 Version (LSW)
21	0014	Battery Voltage
22	0015	Average Motor Current
23	0016	Motor Current Trip Setting
24	0017	Alarm Flags / Alarm Flags 2

## PACS<sup>®</sup> FUNCTION \$41

To convert the Modbus protocol to PACS<sup>®</sup> commands, the OCM-102 employs the user defined **Function Code \$41** (65 decimal). The requesting Modbus *Application Data Unit* (ADU) is shown below:

Address	Function (\$41)	PACS <sup>®</sup> command	CRC or LRC
---------	-----------------	---------------------------	------------

The PACS<sup>®</sup> command consists of one to eight hexadecimal bytes and dictates if any data will be returned in the responding ADU. The responding ADU is shown below:

Address	Function (\$41)	PACS <sup>®</sup> data	CRC or LRC
---------	-----------------	------------------------	------------

If the PACS<sup>®</sup> command does not have any returning data, the CRC or LRC bytes will immediately follow the **Function Code**, \$41. Otherwise, the PACS<sup>®</sup> data will either be 1 byte (SING byte commands), 2 bytes (DOUB byte commands), or 4 bytes (QUAD byte commands). Appendix "A" provides a summary of all PACS<sup>®</sup> commands. Refer to "The PACS<sup>®</sup> Standard" manual for complete details.

The basic structure of the PACS<sup>®</sup> command is shown below:

Command 1 byte	Address 2 bytes	Data 0, 1, 2, or 4 bytes
-------------------	--------------------	-----------------------------

The **Command** is always one byte, referred to as the PACS<sup>®</sup> Code (see Appendix "A"), and dictates how many **Data** bytes, if any, are required.

When required, the **Address** is always a two byte value with the most significant byte following the command byte. Some commands do not require an address, such as Indexed commands; therefore, the data bytes, if any, will follow the command byte.

The number of **Data** bytes is dictated by the command byte. For DOUB (2 bytes) and QUAD (4 bytes) byte values, the most significant byte is sent first. Some commands, such as the INCR (increment) command, do not require data. Other commands, such as the READ command, do not require data bytes in the PACS<sup>®</sup> command, but rather dictates how many bytes are to be returned by the DHC controller. **Data** is returned most significant byte first.

## PACS<sup>®</sup> PARAMETERS

While any two byte address value is valid, only a select few correspond to parameters of significant meaning to the user - these are referred to as the *PACS<sup>®</sup> Parameters*. PACS<sup>®</sup> parameters can either be *Read Only Parameters* or *Read/Write Parameters* - see Sections III and IV for

complete details. Table 2 provides a summary of these parameters.

Reading locations, or addresses, not assigned to a PACS<sup>®</sup> parameter will return data of unknown values. Changing locations that are not assigned as *Read/Write Parameters* are prohibited by the DHC controller and will not alter the contents of such locations. It should be noted that the DHC controller's internal memory pointer, which is used for Indexed commands, will be set according to the PACS<sup>®</sup> command and the provided address; even if the address is not assigned to a parameter.

**Table 2 - DHC Series PACS<sup>®</sup> Parameters**

ADDRESS	SIZE	READ ONLY PARAMETERS
00B0	DOUB	Analog Command Input
00B2	DOUB	Battery Voltage
00B4	DOUB	Average Motor Current
00B6	DOUB	Motor Current Trip Setting
00B8	DOUB	Position
00BD	SING	Alarm Flags 2
00BE	DOUB	Average % Duty Cycle
00C0	DOUB	Maximum Deviation
00C2	QUAD	Hours of Operation
00C6	QUAD	Cycles of Operation
00CA	SING	Alarm Flags
00CB	SING	Mode
00CC	SING	Configuration
00CD	SING	Override Control
606C	DOUB	Serial No. MSB / Model Code
606E	DOUB	Serial Number

ADDRESS	SIZE	READ/WRITE PARAMETERS
00D0	DOUB	Aux Closed Position
00D2	DOUB	Aux Open Position
00D8	DOUB	PACS <sup>®</sup> Command Input
00ED	SING	PACS <sup>®</sup> Control Register
00EE	SING	PACS <sup>®</sup> Timeout Setting
00EF	SING	PACS <sup>®</sup> Reset
6070	DOUB	User Non-volatile Memory
6072	DOUB	
6074	DOUB	
6076	DOUB	
6078	DOUB	
607A	DOUB	
607C	DOUB	
607E	DOUB	



## SECTION II

### Modbus Registers

Modbus Function \$03 can be used to read the parameters described in this section. Registers 1-24 (Register Addresses 0000 to 0017<sub>hex</sub>) are readable parameters. An error response with exception code \$03 is returned if the **Starting Register Address** is greater than 0017<sub>hex</sub>, or if the **Number of Registers** requested results in a **Register Address** that is greater than 0017<sub>hex</sub>.

Modbus Function \$06 can be used to write Registers 1 to 6 (addresses 0000 to 0005). Requesting a register greater than 6 (address 0005) invokes an error response with an exception code of \$03.

<b>OCM-102 PACS® OFF LINE TIMER VALUE</b>	
REGISTER	ADDRESS (hex)
<b>1</b>	<b>0000</b>

The OCM-102 PACS® Master port communicates to the DHC controller's PACS® Slave port through the Option Module Connector. The OCM-102 maintains an off line timer that allows the DHC controller a limited amount of time to respond. If the OCM-102 detects an off line time out, it will return an error response with exception code \$0B. Normally an off line time out occurs when power is lost to the DHC controller. Since the OCM-102 is powered by the DHC controller, it will not provide a respond.

Intermittent power drop outs can also cause an off line time out condition. In this case, the OCM-102 will

return an error response with exception code \$0B. Subsequent requests from the bus will also return an error response until the OCM-102 corrects the detected PACS® error.

The most significant byte of the data is always zero, and the least significant byte represents the PACS® Master off line time out period in 0.1 second increments. So, a value of 26 (\$1A) indicates a time out period of 2.6 seconds. On power up, the OCM-102 automatically sets the Off Line Timer Value to 2 (0.2 seconds).

By design, the 0.2 second timeout period is sufficient for a DHC controller. However, the Off Line Timer Value can be changed to another value. Note that the changed value is lost if the OCM-102 loses power, and the value will need to be set again after power returns. If the most significant byte is not zero when writing the value, an error response with exception code \$03 is returned.

<b>AUX CLOSED POSITION</b>	
REGISTER	ADDRESS (hex)
<b>2</b>	<b>0001</b>

The **Aux Closed Position** parameter is a 16 bit value that represents the *aux close* setting associated with the "close limit alarm" (bit 7 of the **Alarm Flags**, see Registers 15 and 24). The value ranges from 0 to 10000 and represents a setting of 0% to 100%, where 0% is the *closed* position and 100% is the *open* position. For example, read-

ing or writing a value of 2535 (09E7<sub>hex</sub>) represents a setting of 25.35% *open*.

This parameter can be read or written at any time and is stored in non-volatile memory within the DHC controller. When writing values greater than 10000 (2710<sub>hex</sub>), the DHC automatically limits the value to 10000.

The DHC controller will set bit 7 of the **Alarm Flags** parameter to "1" whenever the **Position** parameter (Register 8) is less than the *aux close* setting. Bit 7 is set to "0" when the **Position** parameter is greater than or equal to the *aux close* setting plus 1%. This provides hysteresis that prevents the alarm flag from toggling between "1" and "0" due to random electrical or mechanical variations. In the previous example, the parameter is set to 25.35%. Bit 7 is set to "1" when **Position** is 25.34% or less, and bit 7 is set to "0" when **Position** is 26.35% or more.

It should be noted that the *aux close* position can also be set from the DHC control panel using the AUX CLOSE OUTPUT mode. However, setting this parameter precisely from the DHC controller can be difficult. If a precise setting via the bus is desired, occasionally reading this parameter to insure that an operator has not changed the setting may be prudent.

AUX OPEN POSITION	
REGISTER	ADDRESS (hex)
3	0002

The **Aux Open Position** parameter is used the same way as the **Aux Closed Position** parameter. It is a 16 bit value that represents the *aux open* setting associated with the "open limit alarm" (bit 6 of the **Alarm Flags** parameter). Bit 6 of the **Alarm Flags** parameter is set to "1" whenever the **Position** parameter is greater than the *aux open* setting. Bit 6 is set to "0" when the **Position** parameter is less than or equal to the *aux open* setting minus 1%. The *aux open* position can also be set from the DHC control panel using the AUX OPEN OUTPUT mode.

For example, if the parameter is changed via the bus to 7465 (1D29<sub>hex</sub>), this represents a setting of 74.65%. Bit 6 of **Alarm Flags** is set to "1" when **Position** is 74.66% or more, and bit 6 is set to "0" when **Position** is 73.65% or less.

PACS® COMMAND INPUT	
REGISTER	ADDRESS (hex)
4	0003

The **PACS® (Digital) Command Input** parameter is a 16 bit value that is used to control the position of the valve via the bus. The value ranges from 0 to 10000 and represents 0% to 100% open. For example, reading or writing a value of 4575 (11DF<sub>hex</sub>) represents a command signal

of 45.75% open. When writing values greater than 10000, the DHC automatically limits the value to 10000.

The DHC controller will use the **PACS® Command Input** parameter instead of the **Analog Command Input** parameter (Register 7) when it is configured to do so. Before the DHC uses the **PACS® Command Input** parameter, the **PACS® Timeout Setting** parameter (Register 6) must be set via the bus. As a convenience to the user, writing the **PACS® Command Input** parameter also automatically writes the **PACS® Timeout Setting** parameter. If the **PACS® Timeout Setting** was zero when the **PACS® Command Input** is written, then the **PACS® Timeout Setting** is written to a value of 4 (representing 2 sec). If the **PACS® Timeout Setting** was not zero when the **PACS® Command Input** is written, then the previous timeout setting is not changed.

The DHC can be configured in two ways to use the **PACS® Command Input** parameter. First, the DHC can be configured for a Digital command type from the DHC control panel using the COMMAND TYPE mode. In this case, the DHC controller ignores the analog input signals and will wait for a command from the bus, being the **PACS® Command Input** parameter.

In the second configuration, the DHC is configured for one of the analog command types, and the bus can override the analog command. In this case, the **PACS® Control Register** parameter (Register 5) must also be configured, via the bus, to override the analog signal.

The **PACS® Command Input**, **PACS® Timeout Setting**, and **PACS® Control Register** parameters are not saved in non-volatile memory. This means that the parameters' values are lost when the DHC controller loses power. Following a power up or reset condition, the DHC sets these parameters to zero. Therefore, it may be desirable to read the **PACS® Timeout Setting** regularly to detect when a power down condition has occurred.

PACS® CONTROL REGISTER	
REGISTER	ADDRESS (hex)
5	0004

When reading the **PACS® Control Register** parameter, the most significant byte of the returned data will be 0. The meaning of each bit of the least significant byte is described in the following. The bit assignments are shown below:

```

7 6 5 4   3 2 1 0
                X__command override control
                XX__PACS® manual control
                X___unassigned
                XX_____set loss of command
                X_____unassigned
                X_____power monitor

```

Writing this parameter can only be used to set or clear Bit 0, which is used to override an analog command signal. The remaining bits can only be written using PACS<sup>®</sup> Function \$41 (see Section IV). Writing a value of 00FF<sub>hex</sub> sets Bit 0 to "1", while writing a value of 0000 clears Bit 0 to "0". To prevent an unknown command value from taking effect, the **PACS<sup>®</sup> (Digital) Command Input** parameter (Register 4), should be written before writing Bit 0 to "1".

Following a power up or reset condition, the **PACS<sup>®</sup> Control Register** parameter is cleared to 0. If a PACS<sup>®</sup> timeout occurs, bits 0, 1, and 2 will be cleared by the DHC controller. A PACS<sup>®</sup> timeout occurs when the DHC controller does not receive a requesting ADU (which can be a broadcast command) from the bus after a certain amount of time. The time allowed is set using the **PACS<sup>®</sup> Timeout Setting** parameter (Register 6).

As a convenience to the user, writing the **PACS<sup>®</sup> Control Register** parameter also automatically writes the **PACS<sup>®</sup> Timeout Setting** parameter. If the setting was zero when the **PACS<sup>®</sup> Control Register** is written, then the **PACS<sup>®</sup> Timeout Setting** is written to a value of 4 (representing 2 sec). If the setting was not zero, then the previous timeout setting is not changed.

Since Bit 0 allows the bus to run the actuator, the DHC controller must be in the AUTO mode (see CONTROL HIERARCHY in Section I) before it overrides the analog command. The value of Bit 0 is not affected if the DHC is put in a mode other than AUTO.

**Bit 0 - "command override control"**

When this bit is "0", the DHC controller will use the **Analog Command Input** parameter (Register 7) to control the valve. When this bit is "1", the DHC will use the **PACS<sup>®</sup> Command Input** parameter to control the valve. If the DHC is configured for a Digital command type, this bit is essentially ignored since there is no analog command signal. In this case, the DHC will use the **PACS<sup>®</sup> Command Input** regardless of the value of bit 0.

**Bits 1 & 2 - "PACS<sup>®</sup> manual control"**

These bits are used to directly control the DHC outputs, and can only be written using PACS<sup>®</sup> Function \$41 (see Section IV).

**Bits 3 & 6 - unassigned**

These bits perform no operation, but can be written using PACS<sup>®</sup> Function \$41 (see Section IV).

**Bits 4 & 5 - "set loss of command"**

These bits allow changing the Loss of Command setting via the bus, and can only be written using PACS<sup>®</sup> Function \$41 (see Section IV).

**Bit 7 - "power monitor"**

This bit is used to detect if the DHC experienced a power loss, and can only be written using PACS<sup>®</sup> Function \$41 (see Section IV).

<b>PACS<sup>®</sup> TIMEOUT SETTING</b>	
REGISTER	ADDRESS (hex)
<b>6</b>	<b>0005</b>

The **PACS<sup>®</sup> Timeout Setting** parameter is used to detect when the bus has stopped accessing the DHC controller and is referred to as "loss of communications". When the DHC is configured for a Digital command type, detecting loss of communications will set Bit 0 of the **Alarm Flags** parameter (Register 15 or 24), and the DHC outputs will be controlled according to the "loss of command setting" (see **Configuration** parameter, Register 15). Additionally, the DHC controller will clear Bits 0, 1, and 2 of the **PACS<sup>®</sup> Control Register** parameter (Register 5). This has the effect of disabling bus control of the DHC outputs.

When reading this parameter, the most significant byte of the returned data will be 0, and the least significant byte will be a value from 0 to 127. The value of the parameter represents time in 0.5 second increments. For example, a value of 20 represents 10 seconds. Loss of communications is detected when a requesting ADU, addressed to the node address as set on the OCM-102, has not been received before this time expires (referred to as a PACS<sup>®</sup> timeout).

Upon receiving a valid ADU, the timer is restarted. If an error response is returned by the OCM-102, the timer will not be restarted. Also, no response from the OCM-102 indicates that the timer was not restarted. This is true when the OCM-102 is in the Listen Only mode. Broadcast commands (node 0) will restart the timer; however, since the OCM-102 is required not to respond to a broadcast command, the bus user will not know for certain if the timer has been restarted. Additionally, Function \$08 commands will not restart the timer.

When writing this parameter, the most significant byte must be zero; if not, an error response with exception code \$03 is returned. If the least significant byte is greater than 127, the DHC automatically limits the value to 127. Since the DHC controller clears the **PACS<sup>®</sup> Timeout Setting** to zero on power up, the user can use this parameter to monitor a power down condition. After writing the **PACS<sup>®</sup> Timeout Setting**, the **PACS<sup>®</sup> (Digital) Command Input** parameter (Register 4), or the **PACS<sup>®</sup> Control Register** parameter (Register 5), reading the **PACS<sup>®</sup> Timeout Setting** should return the previously set value. If a value of zero is returned, the user will know that the DHC experienced a power down and up sequence.

## ANALOG COMMAND INPUT

REGISTER	ADDRESS (hex)
<b>7</b>	<b>0006</b>

The DHC controller can be configured to control an automated valve using an analog signal, such as 4-20mA. The **Analog Command Input** parameter is a 16 bit value ranging from 0 to 10000 and represents 0% to 100% open. For example, reading a value of 3486 (0D9E<sub>hex</sub>) represents a value of 34.86% open. The DHC automatically scales this parameter according to the type of analog signal selected:

Analog Input	0% Reading	100% Reading
4-20mA	4mA	20mA
1-5V	1V	5V
2-10V	2V	10V (DHC-400 only)
0-5V	0V	5V
0-10V	0V	10V

If the analog signal is not connected to the DHC controller, a reading of 0% is returned. A reading of 0% is also returned when the analog signal is less than 4mA for a 4-20mA signal, less than 1V for a 1-5V signal, and less than 2V for a 2-10V signal. If the analog input signal exceeds the value associated with 100%, a reading of 100% is returned. If the DHC is configured for a Digital command type, a reading of 0% is returned. A digital command input must come from the bus, and can be read and written using the **PACS® Command Input** parameter (Register 4).

The command input type cannot be selected via the bus, and must be configured at the DHC using the COMMAND TYPE mode. However, the setting can be read using the **Configuration** parameter (Register 15).

## POSITION

REGISTER	ADDRESS (hex)
<b>8</b>	<b>0007</b>

In order to position an automated valve to a desired position (according to the analog or digital command input), the DHC controller uses a feedback potentiometer to monitor the position of the actuator's output shaft that connects to the valve stem. The **Position** parameter is a 16 bit value that is automatically scaled from 0 to 10000, which represents 0% to 100% open. For example, reading a value of 3478 (0D96<sub>hex</sub>) represents a value of 34.78% open.

When setting up the DHC controller, the user can set the closed position, using the CLOSE mode, to any point within the range of the actuator. Likewise, the open position can be set to any desired point using the OPEN mode. The DHC automatically scales the **Position** parameter to associate 0% with the *closed* position, and 100% with the *open* position.

If the feedback potentiometer is operated outside of its range (usually due to incorrect installation), or if it incurs a failure (broken connection, shorted connection, etc.), the **Position** parameter will either read 0% or 100%. When this occurs, the FAULT light on the DHC controller will flash and the outputs to the actuator are turned off. The automation system can monitor this fault condition using the **Alarm Flags** parameter (Register 15 or 24).

## AVERAGE % DUTY CYCLE

REGISTER	ADDRESS (hex)
<b>9</b>	<b>0008</b>

Duty cycle is a measure of the amount of time an actuator is turned on versus the amount of time it is turned off. The measurement is expressed as a percentage of time the actuator is turned on. For example, if over a 50 second period, the actuator was turned on for a total of 10 seconds, it is said that the percent duty cycle is 20% (10 seconds divided by 50 seconds).

In a typical PID control application, the average duty cycle will range from 10% to 20%. Persistent readings above 20% duty can indicate that PID settings are changing too fast for the process being controlled. It can also indicate that the actuator/valve combination is not suitably sized for the application.

To protect an electric actuator motor from overheating, the DHC-100, 300, and 400 Series controllers incorporate a duty control feature that intentionally reduces the duty cycle to a safe level when they detect excessive heating of the motor. This has the effect of slowing down the actuator, which offsets PID signals that are changing too fast. In this case, the reduced duty cycle is reflected in the **Average % Duty Cycle** parameter.

Reading the **Average % Duty Cycle** parameter returns a value from 0 to 100 that represents 0% to 100% duty as measured during the past 80 to 100 seconds. Since this parameter is limited to a value of 100 (64<sub>hex</sub>), the most significant byte of the returned data is always zero.

## MAXIMUM DEVIATION

REGISTER	ADDRESS (hex)
<b>10</b>	<b>0009</b>

While the command input signal provides the desired position, the actual achievable position will most often be somewhat less or more. The difference between the actual position achieved and the desired position is called *deviation*. The DHC controller automatically determines the best performance that can be attained based on the actuator being controlled.

The **Maximum Deviation** parameter provides a measure of this performance. The parameter is a 16 bit value ranging from 0 to 10000 that represents 0% to 100%.

For example, reading a value of 15 (000F<sub>hex</sub>) indicates a maximum deviation of 0.15%. So, if the command input signal is 40.00%, then the position the DHC controller can achieve will be from 39.85% to 40.15%.

In a typical PID application, the **Maximum Deviation** parameter will be 0.50% or less, and quite commonly will vary between 0.10% and 0.20%. When this parameter exceeds 0.50%, the PID control algorithm will tend to move the actuator more often which will be reflected in the **Average % Duty Cycle** parameter (Register 9).

HOURS OF OPERATION	
REGISTER	ADDRESS (hex)
<b>11</b>	<b>000A</b>
<b>12</b>	<b>000B</b>

The DHC controller maintains an internal "hour meter" which can be read as the **Hours of Operation** parameter. The parameter is a 32 bit value that ranges from 0 to 4,294,967,295 representing the number of hours that power has been applied to the DHC since its time of manufacture. The parameter is set to zero at the time of manufacture and cannot be changed or reset by the user. This parameter can be useful in automating preventative maintenance schedules.

Register 11 is the most significant word, while Register 12 is the least significant word. Since Register 12 may roll over between separate readings of Register 11 and 12, it is prudent to read both registers with the same ADU (by requesting both registers); the OCM-102 insures that Register 11 is the correct value when this is done.

If the registers are read separately, the following sequence should be followed to acquire a correct value:

1. Read Register 11
2. Read Register 12
3. If the value of Register 12 is zero, then read Register 11 again for its correct value.

Alternatively, Register 12 may be used alone as an hour meter ranging from 0 to 65,535 hours.

CYCLES OF OPERATION	
REGISTER	ADDRESS (hex)
<b>13</b>	<b>000C</b>
<b>14</b>	<b>000D</b>

While the **Hours of Operation** parameter (Registers 11 and 12) can provide a measure of how long an actuator/valve has been in service, the wear on the mechanical components in an actuator and valve is better measured by how often the valve is moved. The **Cycles of Operation** parameter is a cycle counter that indicates how many times

the actuator was switched from off to on by the DHC controller. The parameter is set to zero at the time of manufacture and cannot be changed or reset by the user. The parameter is a 32 bit value that ranges from 0 to 4,294,967,295 representing the number of cycles.

Register 13 is the most significant word, while Register 14 is the least significant word. Since Register 14 may roll over between separate readings of Register 13 and 14, it is prudent to read both registers with the same ADU (by requesting both registers); the OCM-102 insures that Register 13 is the correct value when this is done.

If the registers are read separately, the following sequence should be followed to acquire a correct value:

1. Read Register 13
2. Read Register 14
3. If the value of Register 14 is zero, then read Register 13 again for its correct value.

Alternatively, Register 14 may be used alone as a cycle counter ranging from 0 to 65,535 cycles.

CONFIGURATION / ALARM FLAGS	
REGISTER	ADDRESS (hex)
<b>15</b>	<b>000E</b>

This register combines two 8-bit parameters, where the most significant byte returned is the **Configuration** parameter, and the least significant byte is the **Alarm Flags** parameter (see Register 24 for details).

The individual bits of the **Configuration** parameter are assigned to various DHC controller settings as shown below, and a description of each bit follows.

```

7 6 5 4   3 2 1 0
                X_polarity
                XXX__command type setting
                XX_____loss of command setting
                XX_____indeterminate

```

#### Bit 0 - "polarity"

This bit is automatically set by the DHC controller's polarity detection feature, and is not meaningful to the automation system except for advanced trouble shooting purposes. When this bit is "0", it indicates that the DHC's M1 output drives the actuator in a direction that corresponds to a decreasing voltage on the DHC's feedback potentiometer WIPER input. The M1 output is the one that turns on with the ▲ button when the DHC controller is in the MANUAL mode. When this bit is "1", it indicates that the M2 output (the ▼ button) corresponds to a decreasing voltage on the WIPER.

### Bits 1, 2, & 3 - "command type setting"

These three bits indicate the command type the DHC controller is expecting to use to position the actuator in the AUTO mode. Only the values shown below are returned.

<b>3 2 1</b>	
0 0 0	Digital command (PACS® Command)
0 0 1	4-20mA command
0 1 0	1-5V command
0 1 1	2-10V command (DHC-400 only)
1 0 0	0-10V command
1 1 0	0-5V command

### Bits 4 & 5 - "loss of command setting"

These bits indicate what action the DHC controller will take when it detects a loss of command. Only the values shown below are returned.

<b>5 4</b>	
0 0	last position (DHC outputs disabled)
0 1	close (DHC moves to <i>closed</i> position)
1 0	open (DHC moves to <i>open</i> position)

**Bits 6 & 7** - These bits are not used by the DHC controller and should be ignored. Their value is unknown, and therefore they are indeterminate.

OVERRIDE CONTROL / MODE	
REGISTER	ADDRESS (hex)
<b>16</b>	<b>000F</b>

This register combines two 8-bit parameters, where the most significant byte returned is the **Override Control** parameter, and the least significant byte is the **Mode** parameter.

When bit 7 of the **Override Control** parameter is "1", then *override operation* is enabled by an external local/remote station. While in *override operation*, the DHC controller is disabled (including the three control panel buttons), and the local/remote station has direct control of the DHC controller's outputs. When bit 7 is "0", the DHC controller is in *normal operation*. The remaining bit assignments are shown below.

7 6 5 4	3 2 1 0
	XX__local/remote function
0 0 0	0 0____unassigned (always "0")
X_____	_____override status

### Bits 0 & 1 - "local/remote function"

These bits are useful for trouble shooting problems with the local/remote switches and indicate which

DHC output is turned on. Only the values shown below are returned.

<b>1 0</b>	
0 0	__off (both outputs off)
0 1	__open output on
1 0	__close output on

**Bits 2, 3, 4, 5, & 6** - UNASSIGNED (always "0")

### Bit 7 - "override status"

A "1" indicates that *override operation* is enabled, and a "0" indicates that the DHC is in *normal operation*. When this bit is "0", the remaining bits will also be "0".

The **Mode** parameter indicates what mode of operation the DHC controller is in. Since the bus cannot control the DHC outputs to the actuator unless the DHC controller is set to the AUTO mode, it can be useful to monitor this parameter. The values of the **Mode** parameter are listed below, and are the only values returned by the DHC controller.

decimal value	DHC mode
0	AUTO
1	MANUAL
2	CLOSE
3	OPEN
4	COMMAND TYPE
5	AUX CLOSE OUTPUT
6	AUX OPEN OUTPUT
7	LOSS OF COMMAND
10	PACS® MANUAL
11	POSITION OUT CAL (Close)
12	POSITION OUT CAL (Open)

When *override operation* is enabled, the **Mode** parameter reports the mode of operation the DHC was in prior to override control. If bit 7 of the **Override Control** parameter is "0", the **Mode** parameter indicates the active mode of operation. If bit 7 of the **Override Control** parameter is "1", the mode of operation indicated by the **Mode** parameter is not active, and the DHC is in *override operation*.

MODEL CODE / SERIAL NUMBER	
REGISTER	ADDRESS (hex)
<b>17</b>	<b>0010</b>
<b>18</b>	<b>0011</b>

A given DHC controller has a *model number* and a *serial number*. Each DHC model has its own serial number sequence, so two different model numbers could have the

same serial number, the combination of the model number and serial number is unique to that specific DHC controller.

The **Serial Number** parameter is a 24 bit value where the most significant byte of Register 17 is the most significant byte of the *serial number*. Register 18 represents the two least significant bytes of the *serial number*. If the 24 bit serial number value is 0, it could indicate that there is an internal problem with the DHC controller. Consult Peaktronics, Inc. for assistance in this case.

The least significant byte of Register 17 is the **Model Code** parameter. Its value indicates which model of DHC controller (see page 1) is being used. The table below shows which *model number* the returned value represents.

Model Code (decimal)	DHC model
1	DHC-100
2	DHC-100A
3	DHC-100B
4	DHC-300
5	DHC-300A
6	DHC-300B
7	DHC-200
8	DHC-200A
9	DHC-200B
10	DHC-100C
11	DHC-100D
12	DHC-100E
13	DHC-100Z (custom product)
14	DHC-200C
15	DHC-200D
16	DHC-200E
17	DHC-400

Note that Model code 13 is assigned to a custom product. A model code of 0 or 255 (FF<sub>hex</sub>) could indicate an internal problem with the DHC controller. Consult Peaktronics, Inc. for assistance in these cases.

<b>OCM-102 VERSION</b>	
REGISTER	ADDRESS (hex)
<b>19</b>	<b>0012</b>
<b>20</b>	<b>0013</b>

These two registers form a 32-bit parameter that allows the user to identify the upgrade level of the OCM-102 being used. Additional parameters may be added in future upgrades to accommodate future DHC controller products. The Version parameter will assist in identifying

whether the OCM-102 being used can access the additional parameters. Contact Peaktronics, Inc. for upgrade information and services.

<b>BATTERY VOLTAGE</b>	
REGISTER	ADDRESS (hex)
<b>21</b>	<b>0014</b>

(DHC-400 only)

The DHC-400 controller monitors the **Battery Voltage**, or the DC power source, that is used to power the unit and the actuator's motor. This parameter is a 16 bit value that is scaled from 0 to 3200 which represents 0 to 32.00 volts. The DHC-400 detects when a low battery voltage or an over voltage condition occurs and provides a low battery voltage or over voltage alarm (see **Alarm Flags 2** parameter, Register 24). For models other than a DHC-400, this parameter will always read 0.

<b>AVERAGE MOTOR CURRENT</b>	
REGISTER	ADDRESS (hex)
<b>22</b>	<b>0015</b>

(DHC-400 only)

The **Average Motor Current** parameter is a measure of the average current drawn by the motor during the past 3 to 4 minutes. This parameter is useful in monitoring the drain on the DC power source; particular to batteries, this parameter can be used to project how long a given battery charge might last. This parameter is a 32 bit value that is scaled from 0 to 2500 that represents 0 to 25.00A. For models other than a DHC-400, this parameter will always read 0.

<b>MOTOR CURRENT TRIP SETTING</b>	
REGISTER	ADDRESS (hex)
<b>23</b>	<b>0016</b>

(DHC-400 only)

The DHC-400 features a current trip function that can be used in place of, or in addition to, torque switches that are commonly used in actuators. The trip current is set by adjusting the on-board trim pot. The setting cannot be overwritten, but can be read. The **Trip Current Setting** parameter is a 16 bit value that is scaled from 0 to 1200 that represents 0 to 12.00A. For models other than a DHC-400, this parameter will always read 0.

ALARM FLAGS / ALARM FLAGS 2	
REGISTER	ADDRESS (hex)
24	0017

This register combines two 8-bit parameters, where the most significant byte returned is the **Alarm Flags** parameter, which can also be read with Register 15. The least significant byte is the **Alarm Flags 2** parameter.

Each bit of the **Alarm Flags** parameter corresponds to one of the alarm conditions monitored by the DHC controller. A bit value of "1" indicates that the associated alarm condition has been detected, while a "0" bit indicates that no alarm condition has been detected. The bit assignment for each alarm condition is shown below, and a description of each bit follows.

7	6	5	4	3	2	1	0
				X			
					X		
				X			
			X				
		X					
	X						
X							
		X					

NOTE: Bits 0-5 are alarm conditions that cause a fault, where the DHC controller disables the outputs to the actuator. A FAULT condition is also indicated on the DHC control panel.

**Bit 0 - "loss of command"**

When configured for a 4-20mA, 1-5V, 2-10V (DHC-400 only), or Digital command type, the DHC controller can detect when the command signal has been lost and sets this bit accordingly. For a 0-5V or 0-10V command type, a loss of command cannot be detected and this bit will always read "0". The DHC can also be configured to run to the *open* or *closed* position when a "loss of command" alarm is detected.

The DHC-100, 200, and 300 Series controllers detect a loss of command when the analog input signal goes below its minimum value (e.g., less than 4mA for 4-20mA command type) or above its maximum value (e.g., more than 20mA for a 4-20mA command type). The DHC-400 differs in that it detects a *command out of range* alarm for signals that are above the maximum value (see **Alarm Flags 2** parameter bit 0).

**Bit 1 - "feedback potentiometer alarm"**

If the feedback potentiometer, which provides position information to the DHC, is operated outside of its range, bit 1 will be set to "1". The out of range condition is detected when the potentiometer reaches less than 5%, or more than 95%, of its resistance value. A broken or shorted wire will cause the feedback signal to be at one of the extremes, and therefore is detected as well.

**Bit 2 - "motor alarm (M1 stall)"**

When the DHC controller turns on its M1 output to run the actuator and does not detect any motion after a period of time, bit 2 will be set to "1". This indicates that the DHC has detected a stall in the M1 direction. Since the DHC-400 measures motor current, this bit is set if the current was greater than 0.5A, otherwise bit 2 of the **Alarm Flags 2** parameter is set. Whether the M1 direction is toward *open* or *closed* depends on where those positions have been set. When the DHC is operated in the manual mode from its control panel, the ▲ button turns on the M1 output.

**Bit 3 - "motor alarm (M2 stall)"**

Bit 3 is set to "1" if the DHC detects a stall in the M2 direction. The M2 direction is associated with the ▼ button when the DHC is operated in the manual mode.

NOTE: When bits 2 and 3 are both set to "1", this indicates a stall detected in both directions. The alarms can be cleared mechanically at the actuator, or by powering the unit down and back up. This allows the DHC to attempt moving the actuator.

**Bit 4 - "open limit switch alarm" (DHC-400 only)**

The actuator's limit switches should be set outside of the *open* and *closed* positions so that the DHC controller can accurately position at 0% and 100%. Since the DHC-400 requires the limit switches be connected to the controller, it will detect when the limit switch is reached and sets bit 4 to "1" accordingly. While this causes a fault condition, the unit flashes the OPEN indicator rather than one of the FAULT indicators.

**Bit 5 - "close limit switch alarm" (DHC-400 only)**

Bit 5 performs a similar function as bit 4 for the close limit switch. The CLOSE indicator flashes, rather than one of the FAULT indicators, when the close limit switch is reached.

**Bit 6 - "aux open limit alarm"**

Mechanical auxiliary switches are commonly installed in actuators to provide a signal when the ac-



tuator's position reaches a critical point before full open. Bit 6 can be used to eliminate the need for such a switch. The desired position can be set using the AUX OPEN OUTPUT mode from the DHC control panel. Alternatively, the *aux open* position can be precisely set using the **Aux Open Position** parameter (Register 3). Bit 6 is set to "1" whenever the actuator's position is between the *aux open* position and the *open* position. Technically, this alarm is not considered a fault condition, and the DHC flashes the AUX OPEN OUTPUT light whenever bit 6 is "1".

**Bit 7 - "aux close limit alarm"**

Bit 7 performs a similar function as bit 6 for the *aux close* position. The *aux close* position can be set with the AUX CLOSE OUTPUT mode on the DHC controller, or by using the **Aux Closed Position** parameter (Register 2). Bit 7 is set to "1", and the DHC flashes the AUX CLOSE OUTPUT light, whenever the actuator's position is between the *aux close* position and the *closed* position.

---

Each bit of the **Alarm Flags 2** parameter corresponds to one of the alarm conditions monitored by a DHC-400 controller. A bit value of "1" indicates that the associated alarm condition has been detected, while a "0" bit indicates that no alarm condition has been detected. These alarm conditions cause a fault, where the DHC-400 controller disables the outputs to the actuator and sets the FAULT lights on the control panel accordingly. The bit assignment for each alarm condition is shown below, and a description of each bit follows. For models other than a DHC-400, this parameter will always read 0.

7 6 5 4	3 2 1 0	
		X__command out of range alarm
		0___unassigned (always "0")
		X____motor alarm (no M1 motion)
		X_____motor alarm (no M2 motion)
		X_____over voltage alarm
		X_____motor current trip alarm (M1)
		X_____motor current trip alarm (M2)
		X_____low battery voltage alarm

**Bit 0 - "command out of range alarm"**

The DHC-400 detects when the analog command input signal exceeds its maximum value (e.g., more than 10V for a 2-10V command signal) and bit 0 is set to "1" accordingly; note that the actuator will run to 100% open before the motor is shut off. When the analog input signal polarity is connected backwards for a 0-5V or 0-10V command signal, bit 0 is also set to "1"; note that the actuator will run to 0% open before the motor is shut off. For a

4-20mA, 1-5V, or 2-10V command signal, backward polarity is detected as a *Loss of Command* and sets bit 0 of the **Alarm Flags** parameter instead.

**Bit 1 - UNASSIGNED (always "0")**

**Bit 2 - "motor alarm (no M1 motion)"**

When the DHC-400 turns on its M1 output to run the actuator and does not detect any motion after a period of time, bit 2 will be set to "1" if the motor current was less than 0.5A. If the motor current was greater than 0.5A, a *motor stall* is detected, and bit 2 of the **Alarm Flags** (\$00CA) parameter is set to "1" instead. Whether the M1 direction is toward *open* or *closed* depends on where those positions have been set. When the DHC-400 is operated in the manual mode from its control panel (see Figure 2), the ▲ button turns on the M1 output.

**Bit 3 - "motor alarm (no M2 motion)"**

Bit 3 is set to "1" in the same manner as bit 2 except for the M2 direction. The M2 direction is associated with the ▼ button when the DHC is operated in the manual mode.

NOTE: When bits 2 and 3 are both set to "1", this indicates no motion detected in both directions. The alarms can be cleared mechanically at the actuator, but also by powering the unit down and back up. This allows the DHC-400 to attempt moving the actuator. A power down sequence can also be simulated via the bus using the **PACS® Reset** parameter (see Section IV).

**Bit 4 - "over voltage alarm"**

Bit 4 is set to "1" whenever the DHC-400 detects a battery voltage above 30VDC. While the battery voltage is above 30VDC, the motor operations are suspended, but continue immediately after the voltage returns below 30VDC. Since this type of event can commonly occur from a transient voltage caused by another piece of equipment sharing the DC power source, bit 4 remains set to "1" for 1 sec after the voltage returns below 30VDC; this allows the bus an opportunity to read the bit for short duration transients. Bit 4 will read "0" when the DC power source is at 30V or less. Some earlier DHC-400 units do not have this feature, in which case, bit 4 will always read "0".

NOTE: Frequent or sustained over voltage conditions can damage the DHC-400. Therefore, the cause of such over voltages should be corrected.

**Bit 5** - "motor current trip alarm (M1)"

When the DHC-400 turns on its M1 output to run the actuator and detects a motor current that exceeds the trip current setting, bit 5 will be set to "1". Whether the M1 direction is toward *open* or *closed* depends on where those positions have been set. When the DHC-400 is operated in the manual mode from its control panel (see Figure 2), the ▲ button turns on the M1 output.

**Bit 6** - "motor current trip alarm (M2)"

When the DHC-400 turns on its M2 output to run the actuator and detects a motor current that exceeds the trip current setting, bit 6 will be set to "1". Whether the M2 direction is toward *open* or *closed* depends on where those positions have been set. When the DHC-400 is operated in the manual mode from its control panel (see Figure 2), the ▼ button turns on the M2 output.

NOTE: When bits 5 and 6 are both set to "1", this indicates a current trip in both directions. The alarms can be cleared mechanically at the actuator, but also by powering the unit down and back up. This allows the DHC-400 to attempt moving the actuator. A power down sequence can also be simulated via the bus using the PACS® **Reset** parameter (see Section IV).

**Bit 7** - "low battery voltage alarm"

Bit 7 is set to "1" whenever the DHC-400 detects a battery voltage below 10VDC. While the battery voltage is below 10VDC, communications, as well as motor operations, are suspended, but continue immediately after the voltage returns above 10VDC. For this reason, bit 7 remains set for 3 seconds after the battery voltage returns above 10VDC.

# SECTION III

## PACS<sup>®</sup> Read Only Parameters

### ANALOG COMMAND INPUT

00B0 DOUB

The DHC controller can be configured to control an automated valve using an **Analog Command Input** signal, such as 4-20mA. This parameter can be read as a DOUB byte value as shown below:

PACS<sup>®</sup> Command (RD#B0):  
51 00 B0

Returned Data (example):  
0D 9E

The returned data is a 16 bit value (DOUB byte) ranging from 0 to 10000 and represents a reading of 0% to 100% open. In the example above, \$0D9E (3486 decimal) represents a value of 34.86% open. The DHC automatically scales this parameter according to the type of analog signal selected:

Analog Input	0% Reading	100% Reading
4-20mA	4mA	20mA
1-5V	1V	5V
2-10V	2V	10V (DHC-400 only)
0-5V	0V	5V
0-10V	0V	10V

If the analog signal is not connected to the DHC controller, a reading of 0% is returned. A reading of 0% is also returned when the analog signal is less than 4mA for a 4-20mA signal, and less than 1V for a 1-5V signal. If the analog input signal exceeds the value associated with 100%, a reading of 100% is returned. If the DHC is configured for a Digital command type, a reading of 0% is returned. A digital command input must come from the bus, and can be read and written using the **PACS<sup>®</sup> Command Input** parameter (see Section IV).

The command input type cannot be selected via the bus, and must be configured at the DHC using the COMMAND TYPE mode. However, the setting can be read using the **Configuration** parameter.

### BATTERY VOLTAGE

(DHC-400 only)

00B2 DOUB

The DHC-400 controller monitors the **Battery Voltage**, or the DC power source, that is used to power the unit and the actuator's motor. This parameter is a DOUB byte value that is automatically scaled from 0 to 3200 which represents 0 to 32.00 volts. Internally, the DHC-400 monitors the battery voltage to detect when a low battery voltage condition occurs and provides a low battery voltage alarm (see Alarm Flags 2). In the following example, \$09A0 (2464 decimal) represents a voltage of 24.64V.

PACS<sup>®</sup> Command (RD#B2):

51 00 B2

Returned Data (example):

09 A0

## AVERAGE MOTOR CURRENT

(DHC-400 only)

00B4 DOUB

The **Average Motor Current** parameter is a measure of the average current drawn by the motor during the past 3 to 4 minutes. This parameter is useful in monitoring the drain on the DC power source; particular to batteries, this parameter can be used to project how long a given battery charge might last. This parameter is a DOUB byte value that is scaled from 0 to 2500 representing 0 to 25.00A. In the following example, \$009B (155 decimal) represents a current of 1.55A.

PACS<sup>®</sup> Command (RD#B4):

51 00 B4

Returned Data (example):

00 9B

## MOTOR CURRENT TRIP SETTING

(DHC-400 only)

00B6 DOUB

The DHC-400 features a current trip function that can be used in place of, or in addition to, torque switches that are commonly used in actuators. The trip current is set by adjusting the on-board trim pot. The setting cannot be overwritten, but can be read. The **Trip Current Setting** parameter is a DOUB byte value that is scaled from 0 to 1200 representing 0 to 12.00A. In the following example, \$0140 (320 decimal) represents a current trip setting of 3.20A.

PACS<sup>®</sup> Command (RD#B6):

51 00 B6

Returned Data (example):

01 40

## POSITION

00B8 DOUB

In order to position an automated valve to a desired position (according to the analog or digital command input), the DHC controller uses a feedback potentiometer to monitor the position of the actuator's output shaft that connects to the valve stem. This parameter is a DOUB byte value that is automatically scaled from 0 to 10000 which represents 0% to 100% open. In the following example, \$0D96 (3478 decimal) represents a position of 34.78% open.

PACS<sup>®</sup> Command (RD#B8):

51 00 B8

Returned Data (example):

0D 96

When setting up the DHC controller, the user can set the closed position, using the CLOSE mode, to any point within the range of the actuator. Likewise, the open position can be set to any desired point using the OPEN mode. The DHC automatically scales the **Position** parameter to associate 0% with the *closed* position, and 100% with the *open* position.

If the feedback potentiometer is operated outside of its range (usually due to incorrect installation), or if it incurs a failure (broken connection, shorted connection, etc.), the **Position** parameter will either read 0% or 100%. When this occurs, the FAULT light on the DHC controller will flash and the outputs to the actuator are turned off. The automation system can monitor this fault condition using the **Alarm Flags** parameter.

## ALARM FLAGS 2

(DHC-400 only)

00BD DOUB

The **Alarm Flags 2** parameter is a SING byte value, where each bit corresponds to one of the alarm conditions monitored by a DHC-400 controller. These are in addition to the alarm conditions that are monitored for the **Alarm Flags** parameter. A bit value of "1" indicates that the associated alarm condition has been detected, while a "0" bit indicates that no alarm condition has been detected. These alarm conditions cause a fault, where the DHC-400 controller disables the outputs to the actuator and sets the FAULT lights on the control panel accordingly.

The bit assignment for each alarm condition is shown below:

7	6	5	4	3	2	1	0	
				X				command out of range alarm
				0				unassigned (always "0")
				X				motor alarm (no M1 motion)
				X				motor alarm (no M2 motion)
				X				over voltage alarm
				X				motor current trip alarm (M1)
				X				motor current trip alarm (M2)
				X				low battery voltage alarm

In the following example, \$A0 (1010 0000 binary) indicates that a "low battery voltage alarm" and a "motor current trip alarm (M1)" have been detected.

PACS® Command (RS#BD):  
50 00 BD

Returned Data (example):  
A0

**Bit 0** - "command out of range alarm"

The DHC-400 detects when the analog command input signal exceeds its maximum value (e.g., more than 10V for a 2-10V command signal) and bit 0 is set to "1" accordingly; note that the actuator will run to 100% open before the motor is shut off. When the analog input signal polarity is connected backwards for a 0-5V or 0-10V command signal, bit 0 is also set to "1"; note that the actuator will run to 0% open before the motor is shut off. For a 4-20mA, 1-5V, or 2-10V command signal, backward polarity is detected as a *Loss of Command* and sets bit 0 of the **Alarm Flags** (\$00CA) parameter instead.

**Bit 1** - UNASSIGNED (always "0")

**Bit 2** - "motor alarm (no M1 motion)"

When the DHC-400 turns on its M1 output to run the actuator and does not detect any motion after a period of time, bit 2 will be set to "1" if the motor current was less than 0.5A. If the motor current was greater than 0.5A, a *motor stall* is detected, and bit 2 of the **Alarm Flags** (\$00CA) parameter is set to "1" instead. Whether the M1 direction is toward *open* or *closed* depends on where those positions have been set. When the DHC-400 is operated in the manual mode from its control panel (see Figure 2), the ▲ button turns on the M1 output.

**Bit 3** - "motor alarm (no M2 motion)"

Bit 3 is set to "1" in the same manner as bit 2 except for the M2 direction. The M2 direction is as-

sociated with the ▼ button when the DHC is operated in the manual mode.

NOTE: When bits 2 and 3 are both set to "1", this indicates no motion detected in both directions. The alarms can be cleared mechanically at the actuator, but also by powering the unit down and back up. This allows the DHC-400 to attempt moving the actuator. A power down sequence can also be simulated via the bus using the **PACS® Reset** parameter (see Section IV).

**Bit 4** - "over voltage alarm"

Bit 4 is set to "1" whenever the DHC-400 detects a battery voltage above 30VDC. While the battery voltage is above 30VDC, the motor operations are suspended, but continue immediately after the voltage returns below 30VDC. Since this type of event can commonly occur from a transient voltage caused by another piece of equipment sharing the DC power source, bit 4 remains set to "1" for 1 sec after the voltage returns below 30VDC; this allows the bus an opportunity to read the bit for short duration transients. Bit 4 will read "0" when the DC power source is at 30V or less. Some earlier DHC-400 units do not have this feature, in which case, bit 4 will always read "0".

NOTE: Frequent or sustained over voltage conditions can damage the DHC-400. Therefore, the cause of such over voltages should be corrected.

**Bit 5** - "motor current trip alarm (M1)"

When the DHC-400 turns on its M1 output to run the actuator and detects a motor current that exceeds the trip current setting, bit 5 will be set to "1". Whether the M1 direction is toward *open* or *closed* depends on where those positions have been set. When the DHC-400 is operated in the manual mode from its control panel (see Figure 2), the ▲ button turns on the M1 output.

**Bit 6** - "motor current trip alarm (M2)"

When the DHC-400 turns on its M2 output to run the actuator and detects a motor current that exceeds the trip current setting, bit 6 will be set to "1". Whether the M2 direction is toward *open* or *closed* depends on where those positions have been set. When the DHC-400 is operated in the manual mode from its control panel (see Figure 2), the ▼ button turns on the M2 output.

NOTE: When bits 5 and 6 are both set to "1", this indicates a current trip in both directions. The alarms can be cleared mechanically at the actuator, but also by powering the unit down and back up.

This allows the DHC-400 to attempt moving the actuator. A power down sequence can also be simulated via the bus using the PACS<sup>®</sup> **Reset** parameter (see Section IV).

#### Bit 7 - "low battery voltage alarm"

Bit 7 is set to "1" whenever the DHC-400 detects a battery voltage below 10VDC. While the battery voltage is below 10VDC, communications, as well as motor operations, are suspended, but continue immediately after the voltage returns above 10VDC. For this reason, bit 7 remains set for 3 seconds after the battery voltage returns above 10VDC.

### AVERAGE % DUTY CYCLE

00BE DOUB

Duty cycle is a measure of the amount of time an actuator is turned on versus the amount of time it is turned off. The measurement is expressed as a percentage of time the actuator is turned on. For example, if over a 50 second period, the actuator was turned on for a total of 10 seconds, it is said that the percent duty cycle is 20% (10 seconds divided by 50 seconds).

In a typical PID control application, the average duty cycle will range from 10% to 20%. Persistent readings above 20% duty can indicate that PID settings are changing too fast for the process being controlled. It can also indicate that the actuator/valve combination is not suitably sized for the application.

To protect an electric actuator motor from overheating, the DHC-100, 300, and 400 Series controllers incorporate a duty control feature that intentionally reduces the duty cycle to a safe level when they detect excessive heating of the motor. This has the effect of slowing down the actuator, which offsets PID signals that are changing too fast. In this case, the reduced duty cycle is reflected in the **Average % Duty Cycle** parameter.

The **Average % Duty Cycle** parameter represents the duty cycle measured during the past 80 to 100 seconds. This parameter is assigned to a DOUB byte value that ranges from 0 to 100, representing 0% to 100% duty.

In the following example, \$000D (13 decimal) represents a duty cycle of 13%.

PACS<sup>®</sup> Command (RD#BE):  
51 00 BE

Returned Data (example):  
00 0D

Since the **Average % Duty Cycle** parameter is limited to a value of 100 (\$64 hexadecimal), the most significant byte of the returned data is always zero. This parameter can also be read as a SING byte value as shown below:

PACS<sup>®</sup> Command (RS#BF):  
50 00 BF

Returned Data (example):  
0D

### MAXIMUM DEVIATION

00C0 DOUB

While the command input signal provides the desired position, the actual achievable position will most often be somewhat less or more. The difference between the actual position achieved and the desired position is called *deviation*. The DHC controller automatically determines the best performance that can be attained based on the actuator being controlled.

The **Maximum Deviation** parameter provides a measure of this performance. The parameter is a DOUB byte value ranging from 0 to 10000 that represents 0% to 100%. In the example below, \$000F (15 decimal) indicates a maximum deviation of 0.15%. If the command input signal is 40.00%, then the position the DHC controller can achieve will be from 39.85% to 40.15%.

PACS<sup>®</sup> Command (RD#C0):  
51 00 C0

Returned Data (example):  
00 0F

In a typical PID application, the **Maximum Deviation** parameter will be 0.50% or less, and quite commonly will vary between 0.10% and 0.20%. When this parameter exceeds 0.50%, the PID control algorithm will tend to move the actuator more often which will be reflected in the **Average % Duty Cycle** parameter.

### HOURS OF OPERATION

00C2 QUAD

The DHC controller maintains an internal "hour meter" which can be read as the **Hours of Operation** parameter. The parameter is a QUAD byte value that ranges from 0 to 4,294,967,295 representing the number of hours that power has been applied to the DHC since its time of

manufacture. The parameter is set to zero at the time of manufacture and cannot be changed or reset by the user. The **Hours of Operation** parameter can be useful in automating preventative maintenance schedules. In the following example, \$0000131C (4892 decimal) indicates that the DHC controller has been powered for 4,892 hours.

PACS® Command (RQ#C2):  
52 00 C2

Returned Data (example):  
00 00 13 1C

## CYCLES OF OPERATION

00C6 QUAD

While the **Hours of Operation** parameter can provide a measure of how long an actuator/valve has been in service, the wear on the mechanical components in an actuator and valve is better measured by how often the valve is moved. The **Cycles of Operation** parameter is a cycle counter that indicates how many times the actuator was switched from off to on by the DHC controller. The parameter is set to zero at the time of manufacture and cannot be changed or reset by the user.

This parameter is a QUAD byte value that ranges from 0 to 4,294,967,295 representing the number of cycles. In the following example, \$001A5E06 (1728006 decimal) indicates that the DHC controller turned the actuator on 1,728,006 times.

PACS® Command (RQ#C6):  
52 00 C6

Returned Data (example):  
00 1A 5E 06

## ALARM FLAGS

00CA SING

The **Alarm Flags** parameter is a SING byte value, where each bit corresponds to one of the alarm conditions monitored by the DHC controller. A bit value of "1" indicates that the associated alarm condition has been detected, while a "0" bit indicates that no alarm condition has been detected. Following is the bit assignment for each alarm condition.

7	6	5	4	3	2	1	0
						X	loss of command
						X	feedback potentiometer alarm
					X		motor alarm (M1 stall)
					X		motor alarm (M2 stall)
				X			open limit switch alarm
				X			close limit switch alarm
				X			aux open limit alarm
				X			aux close limit alarm

Bits 0-5 are alarm conditions that cause a fault, where the DHC controller disables the outputs to the actuator. A FAULT condition is also indicated on the DHC control panel (see Figures 1 and 2).

In the following example, \$81 (1000 0001 binary) indicates that an "aux close limit alarm" and a "loss of command" alarm have been detected.

PACS® Command (RS#CA):  
50 00 CA

Returned Data (example):  
81

### Bit 0 - "loss of command"

When configured for a 4-20mA, 1-5V, 2-10V (DHC-400 only), or Digital command type, the DHC controller can detect when the command signal has been lost and sets this bit accordingly. For a 0-5V or 0-10V command type, a loss of command cannot be detected and this bit will always read "0". The DHC can also be configured to run to the *open* or *closed* position when a "loss of command" alarm is detected.

The DHC-100, 200, and 300 Series controllers detect a loss of command when the analog input signal goes below its minimum value (e.g., less than 4mA for 4-20mA command type) or above its maximum value (e.g., more than 20mA for a 4-20mA command type). The DHC-400 differs in that it detects a *command out of range* alarm for signals that are above the maximum value (see **Alarm Flags 2** (\$00BD) parameter).

### Bit 1 - "feedback potentiometer alarm"

If the feedback potentiometer, which provides position information to the DHC, is operated outside of its range, bit 1 will be set to "1". The out of range condition is detected when the potentiometer reaches less than 5%, or more than 95%, of its resistance value. A broken or shorted wire will cause the feedback signal to be at one of the extremes, and therefore is detected as well.

**Bit 2** - "motor alarm (M1 stall)"

When the DHC controller turns on its M1 output to run the actuator and does not detect any motion after a period of time, bit 2 will be set to "1". This indicates that the DHC has detected a stall in the M1 direction. Since the DHC-400 measures motor current, this bit is set if the current was greater than 0.5A, otherwise bit 2 of the **Alarm Flags 2** (\$00BD) parameter is set. Whether the M1 direction is toward *open* or *closed* depends on where those positions have been set. When the DHC is operated in the manual mode from its control panel (see Figure 1), the ▲ button turns on the M1 output.

**Bit 3** - "motor alarm (M2 stall)"

Bit 3 is set to "1" if the DHC detects a stall in the M2 direction. The M2 direction is associated with the ▼ button when the DHC is operated in the manual mode.

NOTE: When bits 2 and 3 are both set to "1", this indicates a stall detected in both directions. The alarms can be cleared mechanically at the actuator, but also by powering the unit down and back up. This allows the DHC to attempt moving the actuator. A power down sequence can also be simulated via the bus using the **PACS® Reset** parameter (see Section IV).

**Bits 4** - "open limit switch alarm" (DHC-400 only)

The actuator's limit switches should be set outside of the *open* and *closed* positions so that the DHC controller can accurately position at 0% and 100%. Since the DHC-400 requires the limit switches be connected to the controller, it will detect when the limit switch is reached and sets bit 4 to "1" accordingly. While this causes a fault condition, the unit flashes the OPEN indicator rather than one of the FAULT indicators.

**Bits 5** - "close limit switch alarm" (DHC-400 only)

Bit 5 performs a similar function as bit 4 for the close limit switch. The CLOSE indicator flashes, rather than one of the FAULT indicators, when the close limit switch is reached.

**Bit 6** - "aux open limit alarm"

Mechanical auxiliary switches are commonly installed in actuators to provide a signal when the actuator's position reaches a critical point before full open. Bit 6 can be used to eliminate the need for such a switch. The desired position can be set using the AUX OPEN OUTPUT mode from the DHC control panel. Alternatively, the *aux open* position can be precisely set using the **Aux Open**

**Position** parameter (see Section IV). Bit 6 is set to "1" whenever the actuator's position is between the *aux open* position and the *open* position. Technically, this alarm is not considered a fault condition, and the DHC flashes the AUX OPEN OUTPUT light whenever bit 6 is "1".

**Bit 7** - "aux close limit alarm"

Bit 7 performs a similar function as bit 6 for the *aux close* position. The *aux close* position can be set with the AUX CLOSE OUTPUT mode on the DHC controller, or by using the **Aux Closed Position** parameter (see Section IV). Bit 7 is set to "1", and the DHC flashes the AUX CLOSE OUTPUT light, whenever the actuator's position is between the *aux close* position and the *closed* position.

**MODE**

00CB SING

The **Mode** parameter is a SING byte value that indicates what mode of operation the DHC controller is in. Since the bus cannot control the DHC outputs to the actuator unless the DHC controller is set to the AUTO mode, it can be useful to monitor this parameter. The values of the **Mode** parameter are listed below, and are the only values returned by the DHC controller.

decimal value	DHC mode
0	AUTO
1	MANUAL
2	CLOSE
3	OPEN
4	COMMAND TYPE
5	AUX CLOSE OUTPUT
6	AUX OPEN OUTPUT
7	LOSS OF COMMAND
10	PACS® MANUAL
11	POSITION OUT CAL (Close)
12	POSITION OUT CAL (Open)

In the following example, \$0B (11 decimal) indicates that the DHC is in the POSITION OUT CAL (Close) mode.

PACS® Command (RS#CB):  
50 00 CB

Returned Data (example):  
0B

DHC controllers feature an override input that allows an operator to manual control the actuator from an external local/remote station. When *override operation* is



enabled, the **Mode** parameter reports the mode of operation the DHC was in prior to override control. If bit 7 of the **Override Control** parameter is "0", the **Mode** parameter indicates the active mode of operation. If bit 7 of the **Override Control** parameter is "1", the mode of operation indicated by the **Mode** parameter is not active, and the DHC is in *override operation*. If a local/remote station is used, both parameters will need to be monitored to determine the DHC controller's mode of operation.

## CONFIGURATION

00CC SING

The **Configuration** parameter is a SING byte value, where groups of bits are assigned to various DHC controller settings. The bit assignments are shown below:

```

7 6 5 4   3 2 1 0
           X__polarity
          XXX__command type setting
         XX___loss of command setting
        XX_____indeterminate

```

In the following example, \$12 (0001 0010 binary) indicates that the DHC controller is configured for a 4-20mA command input and a loss of command setting that will close the actuator upon a loss of command.

PACS<sup>®</sup> Command (RS#CC):  
50 00 CC

Returned Data (example):  
12

### Bit 0 - "polarity"

This bit is automatically set by the DHC controller's polarity detection feature, and is not meaningful to the automation system except for advanced trouble shooting purposes. When this bit is "0", it indicates that the DHC's M1 output drives the actuator in a direction that corresponds to a decreasing voltage on the DHC's feedback potentiometer WIPER input. The M1 output is the one that turns on with the ▲ button when the DHC controller is in the MANUAL mode. When this bit is "1", it indicates that the M2 output (the ▼ button) corresponds to a decreasing voltage on the WIPER input.

### Bits 1, 2, & 3 - "command type setting"

These three bits indicate the command type the DHC controller is expecting to use to position the

actuator in the AUTO mode. Only the values shown below are returned.

```

3 2 1
0 0 0__Digital command (PACS® Command)
0 0 1__4-20mA command
0 1 0__1-5V command
0 1 1__2-10V command (DHC-400 only)
1 0 0__0-10V command
1 1 0__0-5V command

```

### Bits 4 & 5 - "loss of command setting"

These bits indicate what action the DHC controller will take when it detects a loss of command. Only the values shown below are returned.

```

5 4
0 0__last position (DHC outputs disabled)
0 1__close (DHC moves to closed position)
1 0__open (DHC moves to open position)

```

**Bits 6 & 7** - These bits are not used by the DHC controller and should be ignored. Their value is unknown, and therefore they are indeterminate.

## OVERRIDE CONTROL

00CD SING

The **Override Control** parameter is a SING byte value where bit 7 is "1" when *override operation* is enabled by an external local/remote station. While in *override operation*, the DHC controller is disabled (including the three control panel buttons), and the local/remote station has direct control of the DHC controller's outputs. When bit 7 is "0", the DHC controller is in *normal operation*, and the **Mode** parameter indicates the DHC's mode of operation. The remaining bit assignments are shown below.

```

7 6 5 4   3 2 1 0
           XX__local/remote function
          0 0 0 0 0___unassigned (always "0")
         X_____override status

```

In the following example, \$82 (1000 0010 binary) indicates that the DHC is in *override operation* and the DHC output associated with the *closed* position is on.

PACS<sup>®</sup> Command (RS#CD):  
50 00 CD

Returned Data (example):  
82

### Bits 0 & 1 - "local/remote function"

These bits are useful for trouble shooting problems with the local/remote switches and indicate which DHC output is turned on. Only the values shown below are returned.

#### 1 0

0 0\_\_off (both outputs off)

0 1\_\_open output on

1 0\_\_close output on

### Bits 2, 3, 4, 5, & 6 - UNASSIGNED (always "0")

### Bit 7 - "override status"

A "1" indicates that *override operation* is enabled, and a "0" indicates that the DHC is in *normal operation*. When this bit is "0", the remaining bits will also be "0".

## SERIAL NUMBER

606E DOUB

The **Serial Number** parameter is a DOUB byte value that represents the two least significant bytes of the DHC controller's serial number. This parameter can only be read using the READ DOUB byte command with *direct addressing* (PACS® Code \$51) as shown in the example below. Any other READ command will return \$FF data.

PACS® Command (RD#606E):

51 60 6E

Returned Data (example):

05 E3

To obtain the complete serial number, the most significant byte of the **Model Code** parameter provides the most significant byte of the serial number.

## SERIAL No. MSB / MODEL CODE

606C DOUB

The **Model Code** parameter is a DOUB byte value that indicates which model of DHC controller (see page 1) is being used. This parameter can only be read using the READ DOUB byte command with *direct addressing*

(PACS® Code \$51) as shown in the following example. Any other READ command will return \$FF data.

PACS® Command (RD#606C):

51 60 6C

Returned Data (example):

01 02

In the previous example, the most significant byte, \$01, provides the most significant byte of the 3 byte serial number value. The least significant byte, \$02 (2 decimal), represents the DHC model as shown in the following table. Refer to the most recent revision of this manual for model codes that are not shown. No model is assigned a Model Code of \$00 or \$FF (255 decimal). Model code 13 (\$0D) is assigned to a custom product; consult Peaktronics, Inc. for details.

Model Code (decimal)	DHC model
1	DHC-100
2	DHC-100A
3	DHC-100B
4	DHC-300
5	DHC-300A
6	DHC-300B
7	DHC-200
8	DHC-200A
9	DHC-200B
10	DHC-100C
11	DHC-100D
12	DHC-100E
13	DHC-100Z (custom product)
14	DHC-200C
15	DHC-200D
16	DHC-200E
17	DHC-400

Each DHC model has its own serial number sequence, so two different model numbers could have the same serial number. By reading both the **Model Code** parameter and the **Serial Number** parameter, a specific DHC controller can be identified. In the previous examples, model code, \$02, indicates a DHC-100A with serial number 67043 (\$0105E3). This combination is unique to that specific DHC controller.

If the 3 byte serial number value reads \$000000, it could indicate that there is an internal problem with the DHC controller. A model code of \$00 or \$FF (255 decimal) could also indicate an internal problem. Consult Peaktronics, Inc. for assistance in these cases.

## SECTION IV

### PACS<sup>®</sup> Read/Write Parameters

#### AUX CLOSED POSITION

00D0 DOUB

The **Aux Closed Position** parameter is a DOUB byte value that represents the *aux close* setting associated with the "close limit alarm" (bit 7 of the **Alarm Flags** parameter described in Section III). The value ranges from 0 to 10000 and represents a setting of 0% to 100%, where 0% is the *closed* position and 100% is the *open* position. In the following example, the parameter is changed via the bus to \$09E7 (2535 decimal), which is a setting of 25.35%.

PACS<sup>®</sup> Command (CD#D0,2535):

82 00 D0 09 E7

This parameter can be read (as shown below) or written at any time and is stored in non-volatile memory within the DHC controller. The OCM-102 returns a normal response to a CHANGE command, only when the DHC controller has successfully stored the value. When writing values greater than 10000 (2710 hexadecimal), the DHC automatically limits the value to 10000.

PACS<sup>®</sup> Command (RD#00D0):

51 00 D0

Returned Data (example):

09 E7

The DHC controller will set bit 7 of the **Alarm Flags** parameter to "1" whenever the **Position** parameter (see Section III) is less than the *aux close* setting. Bit 7 is set to "0" when the **Position** parameter is greater than or equal to the *aux close* setting plus 1%. This provides hysteresis that prevents the alarm flag from toggling between "1" and "0" due to random electrical or mechanical variations. In the previous example, the parameter is set to 25.35%. Bit 7 is set to "1" when **Position** is 25.34% or less, and bit 7 is set to "0" when **Position** is 26.35% or more.

It should be noted that the *aux close* position can also be set from the DHC control panel using the AUX CLOSE OUTPUT mode. However, setting this parameter precisely from the DHC controller can be difficult. If a precise setting via the bus is desired, occasionally reading this parameter to insure that an operator has not changed the setting may be prudent.

#### AUX OPEN POSITION

00D2 DOUB

The **Aux Open Position** parameter is used the same way as the **Aux Closed Position** parameter. It is a DOUB byte value that represents the *aux open* setting associated with the "open limit alarm" (bit 6 of the **Alarm Flags** parameter). Bit 6 of the **Alarm Flags** parameter is set to "1" whenever the **Position** parameter is greater than

the *aux open* setting. Bit 6 is set to "0" when the **Position** parameter is less than or equal to the *aux open* setting minus 1%. The *aux open* position can also be set from the DHC control panel using the AUX OPEN OUTPUT mode.

In the following example, the parameter is changed via the bus to \$1D29 (7465 decimal), which is a setting of 74.65%. Bit 6 of **Alarm Flags** is set to "1" when **Position** is 74.66% or more, and bit 6 is set to "0" when **Position** is 73.65% or less.

PACS® Command (CD#D2,7465):  
82 00 D2 1D 29

The parameter can be read as shown below:

PACS® Command (RD#00D2):  
51 00 D2

Returned Data (example):  
1D 29

## PACS® COMMAND INPUT

00D8 DOUB

The **PACS® Command Input** parameter is a DOUB byte value that is used to control the position of the valve via the bus. The value ranges from 0 to 10000 and represents 0% to 100% open. The parameter can be written at anytime as shown in the example below. The value of \$11DF (4575 decimal) represents a command signal of 45.75% open. When writing values greater than 10000 (2710 hexadecimal), the DHC automatically limits the value to 10000.

PACS® Command (CD#D8,4575):  
82 00 D8 11 DF

The parameter can be read at anytime as shown below:

PACS® Command (RD#00D8):  
51 00 D8

Returned Data (example):  
11 DF

The DHC controller will use the **PACS® Command Input** parameter instead of the **Analog Command Input** parameter (see Section III) when it is configured to do so. Before the DHC uses the **PACS® Command Input** parameter, the **PACS® Timeout Setting** parameter must be set via the bus. If the timeout setting is zero when the **PACS® Command Input** parameter is written to a DHC-

400 controller, the DHC-400 will automatically set the timeout value to 4 (2 seconds).

The DHC can be configured in two ways to use the **PACS® Command Input** parameter. First, the DHC can be configured for a Digital command type from the DHC control panel using the COMMAND TYPE mode. In this case, the DHC controller ignores the analog input signals and will wait for a command from the bus, being the **PACS® Command Input** parameter.

In the second configuration, the DHC is configured for one of the analog command types, and the bus can override the analog command. In this case, the **PACS® Control Register** parameter must also be configured, via the bus, to override the analog signal.

The **PACS® Command Input**, **PACS® Timeout Setting**, and **PACS® Control Register** parameters are not saved in non-volatile memory. This means that the parameters' values are lost when the DHC controller loses power, or is reset using the **PACS® Reset** parameter. Following a power up or reset condition, the DHC sets these parameters to zero. Therefore, it may be desirable to use the **PACS® Control Register** parameter to detect when a power down condition has occurred.

## PACS® CONTROL REGISTER

00ED SING

The **PACS® Control Register** parameter is a SING byte value, where groups of bits are assigned to various DHC controller functions. The bit assignments are shown below:

7	6	5	4	3	2	1	0	
								X__command override control
								XX__PACS® manual control
								X_____unassigned
								XX_____set loss of command
								X_____unassigned
								X_____power monitor

The parameter can be written at anytime as shown in the following example, where \$80 (1000 0000 binary) sets bit 7 ("power monitor") to "1" and bits 0 to 6 to "0".

PACS® Command (CS#ED,#80):  
62 00 ED 80

The parameter can be read at anytime as shown below:

PACS® Command (RS#00ED):  
50 00 ED

Returned Data (example):  
80

On power up, or following a reset using the **PACS® Reset** parameter, the **PACS® Control Register** parameter is cleared to \$00. If a **PACS®** timeout occurs, bits 0, 1, and 2 will be cleared by the DHC controller. A **PACS®** timeout occurs when the DHC controller does not receive a requesting ADU (which can be a broadcast command) from the bus after a certain amount of time. The time allowed is set using the **PACS® Timeout Setting** parameter.

Note that the **PACS® Timeout Setting** parameter is also cleared on power up and must be set to a non-zero value before bits 0 to 2 can be changed. If the timeout setting is zero when the **PACS® Control Register** parameter is written to a DHC-400 controller, the DHC-400 will automatically set the timeout value to 4 (2 seconds) if bit 0, 1, or 2 are set to "1".

Since bits 0 to 2 allow the bus to run the actuator, the DHC controller must be in the AUTO mode (see CONTROL HIERARCHY in Section I) before it acts on these bits. The values of bits 0 to 2 are not affected if the DHC is put in a mode other than AUTO.

**Bit 0** - "command override control"

When this bit is "0", the DHC controller will use the **Analog Command Input** parameter (see Section III) to control the valve. When this bit is "1", the DHC will use the **PACS® Command Input** parameter to control the valve. If the DHC is configured for a Digital command type, this bit is essentially ignored since there is no analog command signal. In this case, the DHC will use the **PACS® Command Input** regardless of the value of bit 0.

**Bits 1 & 2** - "PACS® manual control"

These bits are used to directly control the DHC outputs in the same way as the MANUAL mode from the DHC control panel. The bits are assigned as follows:

- 2 1**
- 0 0 \_\_PACS® manual control disabled
- 0 1 \_\_turn on M2
- 1 0 \_\_turn on M1
- 1 1 \_\_turn off M1 and M2

When both bits are "0", manual control from the bus is disabled. When either bit is set to "1", the **Mode** parameter (see Section III) will read 10 decimal to indicate **PACS® Manual mode**. Also, bit 0 and the command input signal (analog or digital) are ignored, which gives "PACS® manual control" priority over "command override control". The M1 output is the one that turns on with the ▲ button when the DHC controller is in the MANUAL mode, and the M2 output corresponds to the ▼

button. The direction the actuator moves depends on the wiring to the actuator motor.

CAUTION! These bits turn on the DHC outputs; power to the output is maintained until turned off via the bus. Limit switches in the actuator will be required to disconnect power at the mechanical limits of the actuator. Upon reaching a limit switch, or a mechanical limit, the DHC controller will detect a stall which can be read using the alarm flags parameters (see Section III).

**Bits 3 & 6** - unassigned

Bits 3 and 6 are unassigned and can be set to "1" or "0", but their values perform no function. Using these bits is not recommended since future versions of the DHC Series may assign these bits to other control functions.

**Bits 4 & 5** - "set loss of command"

When configured for a 4-20mA, 1-5V, 2-10V (DHC-400 only), or Digital command type, the DHC controller can detect when the command signal has been lost. From the DHC control panel, the DHC can be configured to go to *close*, go to *open*, or turn off the outputs (last position) using the LOSS OF COMMAND mode. Alternatively, the DHC can be configured via the bus using bits 4 and 5 as shown below:

- 5 4**
- 0 1 \_\_last position (both outputs off)
- 1 0 \_\_close
- 1 1 \_\_open

These bits will always read zero; to read the setting, use the **Configuration** parameter (see Section III). Setting both bits to "0" has no affect, and the previous setting remains. The setting is saved in non-volatile memory, so the setting is retained until it is changed from the DHC control panel or via the bus.

**Bit 7** - "power monitor"

This bit is set to "0" when the DHC powers up, or is reset using the **PACS® Reset** parameter. This bit can only be set to "1" via the bus. After setting this bit to "1", the bit can be monitored to detect that the DHC controller experienced a power loss. This is particularly useful if the bus is being used to control the DHC outputs - see **PACS® Command Input** and **PACS® Timeout Setting** parameters.

Since the **PACS® Control Register** parameter is divided into groups of bits, setting particular bits without affecting others can be more effectively done using the

PACS<sup>®</sup> logic commands. The OR command can be used to set specific bits to "1", while the AND command can be used to clear specific bits to "0". The EX OR (exclusive-or) command can be used to toggle specific bits between "0" and "1". In the following example, the OR command is used to set bit 7, while leaving bits 0 to 6 unchanged.

PACS<sup>®</sup> Command (ORS#ED,#80):  
 6A 00 ED 80

## PACS<sup>®</sup> TIMEOUT SETTING

00EE SING

The **PACS<sup>®</sup> Timeout Setting** parameter is a SING byte value, where all values from \$00 to \$FF are valid. The value of the parameter represents time in 0.5 second increments. For example, a value of \$14 (20 decimal) represents 10 seconds. If a PACS<sup>®</sup> command is not received before this time expires (referred to as a PACS<sup>®</sup> timeout), the DHC controller will clear bits 0, 1, and 2 of the **PACS<sup>®</sup> Control Register** parameter. This has the effect of disabling bus control of the DHC outputs.

The DHC uses a PACS<sup>®</sup> timeout to detect a loss of command when it is configured for a Digital command type. In this case, bit 0 of the **Alarm Flags** parameter (see Section III) is set to "1", and the DHC outputs are controlled according to the "loss of command setting" (see **Configuration** parameter in Section III).

Note that the OCM-102 Modbus Option Module supports other Modbus functions. The OCM-102 only sends PACS<sup>®</sup> commands to the DHC in response to Function Code \$41 (PACS<sup>®</sup> Function). Broadcast commands (node 0) with the PACS<sup>®</sup> Function are processed as normal; however, the OCM-102 will not send a responding ADU.

The **PACS<sup>®</sup> Timeout Setting** parameter can be written at anytime as shown in the example below:

PACS<sup>®</sup> Command (CS#EE,20):  
 62 00 EE 14

The parameter can be read at anytime as shown below:

PACS<sup>®</sup> Command (RS#00EE):  
 50 00 EE

Returned Data (example):  
 14

## PACS<sup>®</sup> RESET

00EF SING

The **PACS<sup>®</sup> Reset** parameter is a SING byte value that is used to force a DHC controller reset. This has the same effect as powering the DHC off and then on. To reset the DHC controller via the bus, write a value of \$FF to this parameter as shown in the following example. Reading this parameter returns an unknown value and has no significant meaning to the user.

PACS<sup>®</sup> Command (CS#EF,#FF):  
 62 00 EF FF

## USER NON-VOLATILE MEMORY

6070 DOUB  
 6072 DOUB  
 6074 DOUB  
 6076 DOUB  
 6078 DOUB  
 607A DOUB  
 607C DOUB  
 607E DOUB

This group of DOUB byte values provides 16 bytes of non-volatile memory that may be used by the user for any purpose. The initial values of these locations are cleared to zero at the time of manufacture. Writing these locations can only be done using the CHANGE DOUB byte command with *direct addressing* (PACS<sup>®</sup> Code \$82) as shown in the example below. Using any other writing command will result in no operation.

PACS<sup>®</sup> Command (CD#6070,#03E8):  
 82 60 70 03 E8

These locations can only be read using the READ DOUB byte command with *direct addressing* (PACS<sup>®</sup> Code \$51) as shown below. Any other READ command will return \$FF data.

PACS<sup>®</sup> Command (RD#6070):  
 51 60 70

Returned Data (example):  
 03 E8

## UNASSIGNED LOCATIONS

00D4 to 00D7     4 bytes

00DA to 00EC     19 bytes

These **Unassigned Locations** can be read or written at any time. Any value written to these locations will be lost when the DHC controller loses power, or is reset using the **PACS<sup>®</sup> Reset** parameter. On power up, or following a **PACS<sup>®</sup>** reset, these locations are cleared, and therefore will always read \$00, unless the user changes their value. Using **Unassigned Locations** is not recommended since future versions of the DHC Series may assign any of these locations for specific control functions.

# SECTION V

## APPLICATION INFORMATION

### BUS TIMING

A Modbus serial line bus consists of one master device and at least one slave device, but not more than 247 slave devices. Each slave device must be configured for a unique node address (1 to 247), while the master device does not have a node address. All communications on the bus are controlled by the master device, where the master transmits a *requesting ADU* addressed to a specific node, or slave device, and then waits for a reply, called the *responding ADU*, from the addressed node.

If the addressed node does not exist on the bus, or the specific slave device encountered a problem (such as power loss, parity error, etc.), no response from the addressed node will be forthcoming. In this case, the master will need to know how much time the specific slave device needs to respond before abandoning the request.

The master device can also send a *broadcast command* to all slave devices by sending a requesting ADU to address zero, where the slave devices execute the requested function, but do not send a responding ADU. In this case, the master device will need to know how much time the slave devices need to process such a request before sending a new requesting ADU.

Referring to Figure 3, the *response time*,  $t_{RESP}$ , is defined as the time from the last **stop bit** of the requesting ADU to the **start bit** of the responding ADU. Response time will vary depending on the Modbus settings and function codes used.

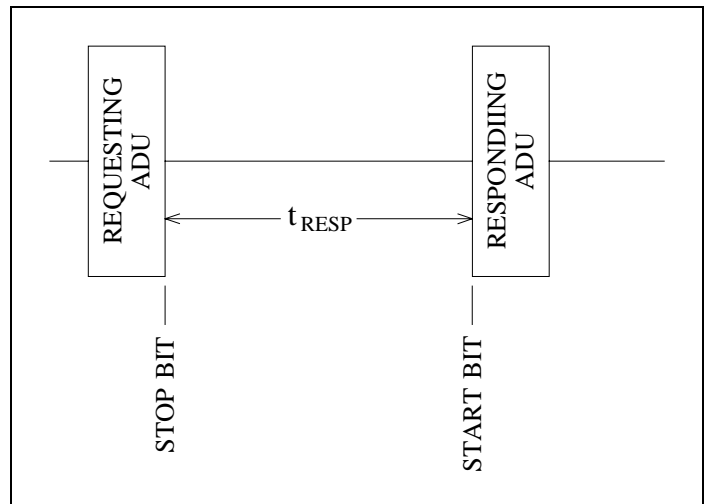


Figure 3 - Response Time

Modbus specifications for the RTU mode require that  $t_{RESP}$  be a minimum of 3.5 character periods. To avoid errors caused by tolerances in the timers in the Modbus master device and the OCM-102, the OCM-102 has a minimum  $t_{RESP}$  of 4.0 character periods. There is no minimum requirement for the ASCII mode; the responding ADU may immediately follow the requesting ADU.

When using PACS<sup>®</sup> Function \$41, the bus is directly communicating with the DHC controller. Since the DHC must perform many other tasks, the response time is more dependent on the DHC controller. Typically,  $t_{RESP}$  will be 2 to 6 msec, unless limited by the minimum 4.0 character periods when using the RTU mode. Certain tasks



and operations of the DHC controller will occasionally require a much greater time to respond, but never more than 200 msec. This is why the **OCM-102 PACS® Off Line Timer Value** (Register 1) defaults to 0.2 seconds on power up. Changing the value to something greater will increase the maximum response time accordingly.

When using Modbus Functions \$03, \$06, and \$08, the OCM-102 buffers the bus from the DHC controller. Therefore the OCM-102 provides a much faster response. When using the ASCII mode, the maximum  $t_{RESP}$  will be 200 usec. In RTU mode, the maximum  $t_{RESP}$  will be 4.0 character periods plus 100 usec.

Response Times Functions \$03, \$06, and \$08		
Baud	RTU mode	ASCII mode
	$t_{RESP}$ max (usec)	$t_{RESP}$ max (usec)
9600	4,683	200
19200	2,392	200
57600	864	200
115200	482	200

Response Times (Function \$41 - any mode or baud)	
$t_{RESP}$ typical	2 to 6 msec
$t_{RESP}$ max	200 msec

## USING A DIGITAL COMMAND TYPE

To control the DHC controller with a command signal from the bus instead of from an analog input, the DHC must first be configured for a Digital command type using the COMMAND TYPE mode; the command input type cannot be selected via the bus. When configured for a Digital command type, the DHC ignores any analog command signal that may be connected, and the **Analog Command Input** parameter (Register 7) will read zero.

On power up, the DHC controller will detect a loss of command until it receives the appropriate bus commands for controlling the actuator. Bit 0 of the **Alarm Flags** parameter (Register 15 or 24), will be set to "1", and the DHC controller will move the valve to the default position (open, closed, or last position) as set by the LOSS OF COMMAND mode from the DHC control panel. Note, the default position cannot be set using Function \$06. The default position can be changed via the bus using Function \$41 to set bits 4 and 5 of the **PACS® Control Register** parameter as desired (see Section IV).

To provide the command signal value to the DHC, the Modbus master device writes the desired position value (0 to 10000, representing 0% to 100% open) to the **PACS® Command Input** parameter (Register 4). However, the DHC controller will not act on the command value until the **PACS® Timeout Setting** parameter (Register 6) is set to a

non-zero value; the parameter is cleared to zero when the DHC powers up. Upon setting the timeout value (1 to 127, representing 0.5 seconds to 63.5 seconds), the DHC immediately clears the loss of command alarm and begins moving the valve to the desired command value.

Since the command input value is also cleared to zero on power up, it makes sense to write the command input value before setting the timeout value. Once the timeout value is set, it is not necessary to rewrite it unless the DHC controller loses power. Note, writing the **PACS® Command Input** parameter with Function \$06 automatically sets the timeout value (see Section II).

Momentary power drop outs are common in industrial applications and could very likely occur between updates of the command value. The bus master may want to know that power was lost, so that it can rewrite the command and/or timeout values when power returns. To monitor for a power drop out, the **PACS® Timeout Setting** parameter can be read using Function \$03 before using Function \$06 to update the command and/or timeout settings. If power was lost, the timeout setting will read zero. A power drop out can also be detected by setting bit 7 of the **PACS® Control Register** parameter (00ED SING) to "1" using PACS® Function \$41. Since a power drop out will cause bit 7 to be cleared to "0", the bus master can detect that a power loss occurred by monitoring bit 7 at whatever interval is suitable for the application.

Based on the previous discussion, consider an application using RTU mode where the bus master scans each node by requesting all 24 available registers with Function \$03. The entire read sequence for a single node would require 8 characters for the Requesting ADU + 4 characters (plus 100 usec) for  $t_{RESP}$  + 53 characters for the Responding ADU + 3.5 characters required idle time for a total read time of 68.5 characters (plus 100 usec). After reading all the parameters, it would be logical that the command input value would then be updated using Function \$06. The write sequence for this would be an additional 23.5 characters (plus 100 usec). Also, if the bus master requires more time than 3.5 characters to prepare the next Requesting ADU, such time will add to the total sequence time. Assuming no additional delay from the bus master, the scan time for each node is shown below.

Scan Time per node (read all 24 parameters + write command)	
Baud	msec
9600	105.63
19200	52.92
57600	17.77
115200	8.94

When writing the **PACS® Command Input** parameter with Function \$06, the **PACS® Timeout Setting** parameter is automatically written for 2 seconds (a value of 4) if it was previously zero. Assuming the application

implements 20 nodes running at 9600 baud, the fastest each node could be accessed would be once every 2.113 seconds (20 times 105.63 msec). Each node would detect a loss of command every 2 seconds - obviously an undesired effect.

One obvious solution to this problem is to run the bus at 19200 baud, thus reducing the access time to 1.058 seconds. However, the cable type and cable lengths may not lend themselves to a higher baud rate. One might also consider a two scan process where 12 registers are read on one pass through all the nodes, and then reading the other 12 registers on a second pass. However, this process is less efficient due to all the additional requesting ADU's.

Another obvious solution might be to write the timeout setting, to say 4 seconds. This however would require the bus master to monitor power dropouts on each node to insure that the timeout setting does not default to 2 seconds. Further, it may not be desirable to have a node delay detection of loss of command for 4 seconds.

A simple solution that avoids possible undesired side effects involves using a broadcast command that writes the **PACS® Timeout Setting** parameter to a desired time. Sending the broadcast command after scanning 10 nodes causes all the nodes to restart their timers, thus allowing time to scan another 10 nodes. Repeating this process, allows adding another 10 nodes to the bus without further concern about the timeout settings. If the bus truly drops out (e.g., a power loss to the bus master device), each node would detect a loss of command within 2 seconds (or whatever time period was set).

## OVERRIDING AN ANALOG COMMAND TYPE

Systems using large numbers of nodes for controlling various processes may be designed to operate the DHC controllers with analog signals. In such a system the Modbus master device might be equipped with a digital to analog interface which can be updated more rapidly than sending a serial sequence of **PACS® Command Input** parameters to a large number of nodes. The digital to analog interface would provide a constant signal to each of the DHC controllers, thus eliminating concerns of a **PACS®** timeout. In this type of system, the master device can "override" the analog command signal by merely setting the appropriate analog output to any desired value. Therefore, the bus is only used to monitor various parameters in the DHC controller, and the **PACS® Timeout Setting** parameter can be ignored.

Due to the complexities of PID process control loops, it might not be practical for the Modbus master device to handle a large number of nodes. In this instance, it is likely that dedicated PID control devices, which might even be nodes on the bus, are used instead. In this type of system, a PID controller provides an analog command signal to a DHC controller associated with the process.

The bus master can override an analog command signal that comes from a dedicated PID controller by setting bit 0 of the **PACS® Control Register** parameter (Register 5) to "1". While bit 0 is "1", the DHC controller will use the **PACS® Command Input** value (Register 4) instead of the analog command signal to control the actuator position. However, before bit 0 can be set, the **PACS® Timeout Setting** parameter (Register 6) must be set to a non-zero value since bit 0 is cleared to "0" when a timeout occurs. Using Function \$06 to write the **PACS® Control Register** parameter (see Section II) automatically sets the **PACS® Timeout Setting** parameter to 4 (2 seconds) if it was previously zero, and then sets bit 0 of the control register.

When a large number of nodes are implemented, the time required to scan all the nodes may take longer than the 2 second timeout setting. To resolve this issue, the same concerns and techniques employed for using a digital command apply here. In the example application discussed in "Using a Digital Command Type", the **PACS® Timeout Setting** parameter is written with a broadcast command after 10 nodes are scanned, and the **PACS® Command Input** parameter is written when the specific node is scanned.

When overriding an analog command that normally controls the DHC controller, it is presumed that the override function is a "temporary" or "special" operation. This suggests that only one, or just a few, DHC controllers are being overridden at a given time. Therefore, it would not make sense to write the **PACS® Command Input** parameter every time a given node is scanned. Instead it would be more sensible to write the command value only when the "special" event occurs that requires overriding the analog command. The first effect of doing this reduces the scan time per node as shown below.

<b>Scan Time per node</b> (read all 24 parameters)	
<b>Baud</b>	<b>msec</b>
9600	78.60
19200	39.35
57600	13.18
115200	6.61

At 9600 baud, scanning 10 nodes would require 0.786 seconds. This allows sufficient time to refresh the command input values and the **PACS® Control Register** parameters of the specific nodes being overridden.

Keep in mind that this application could likely have PID controllers that are nodes on the bus as well. The time required to scan or access these devices will need to be accounted for. Additionally, using broadcast commands to refresh the timeout setting may not be practical since a PID controller may respond differently to such a command.

If a DHC controller experiences a power drop out, the command input value will be cleared to zero. Also, bit 0 of the **PACS® Control Register** parameter is cleared,

which in turn terminates override control. This is why the command value and the control register are refreshed after each scan of 10 nodes.

Given the foregoing, an example sequence for overriding the analog command signal for node 5 would be as follows:

- 1) Scan 10 nodes, reading all 24 parameters in each node. This requires 0.786 seconds.
- 2) Use Function \$06 to write the **PACS<sup>®</sup> Command Input** parameter (Register 4) of node 5 with the value to override the analog command. Note, a value of 0 to 10000 represents 0% to 100% *open*. This also automatically writes the **PACS<sup>®</sup> Timeout Setting** parameter (Register 6) for a 2 second timeout if it was not previously set, or if a power drop out occurred.
- 3) Use Function \$06 to write the **PACS<sup>®</sup> Control Register** parameter (Register 5) of node 5 with a value of 00FF<sub>hex</sub>. This sets bit 0 of the control register to "1" which commands the DHC controller to use the override command value written in step 2 instead of the analog command input signal.
- 4) Repeat these steps, scanning the next 10 nodes. Note, the nodes could also be PID controllers provided that the total time required for the 10 nodes is less than 2 seconds.

In this example, if node 5 had a power drop out, override control from the bus would be restored within 1 second after power returned. If the bus dropped out (e.g., a power loss to the bus master device), override control of node 5 would be terminated within 2 seconds, and node 5 would resume using its analog command input signal.

# APPENDIX A

## PACS<sup>®</sup> Reference Guide

	PACS <sup>®</sup> CODE	ASCII	DATA MODE	ADDR MODE		PACS <sup>®</sup> CODE	ASCII	DATA MODE	ADDR MODE
<b>A</b> ADD	64	ASa,d	SING	DIR	<b>L</b> LEVEL	1C	L	SING	
	84	ADa,d	DOUB	DIR					
	C4	AQa,d	QUAD	DIR					
	25	AS,d	SING	IND					
	45	AD,d	DOUB	IND					
	85	AQ,d	QUAD	IND	<b>OR</b> OR	6A	ORSa,d	SING	DIR
<b>AN</b> AND	68	ANSa,d	SING	DIR		8A	ORDa,d	DOUB	DIR
	88	ANDa,d	DOUB	DIR		CA	ORQa,d	QUAD	DIR
	C8	ANQa,d	QUAD	DIR		2B	ORS,d	SING	IND
	29	ANS,d	SING	IND		4B	ORD,d	DOUB	IND
	49	AND,d	DOUB	IND	8B	ORQ,d	QUAD	IND	
	89	ANQ,d	QUAD	IND	<b>R</b> READ	50	RSa,#	SING	DIR
<b>C</b> CHANGE	62	CSa,d	SING	DIR		51	R#a,#	DOUB	DIR
	82	CDa,d	DOUB	DIR		52	RQa,#	QUAD	DIR
	C2	CQa,d	QUAD	DIR		10	RS,#	SING	IND
	23	CS,d	SING	IND		11	R#,#	DOUB	IND
	43	CD,d	DOUB	IND	12	RQ,#	QUAD	IND	
	83	CQ,d	QUAD	IND	<b>S</b> SUB	66	SSa,d	SING	DIR
<b>D</b> DECR	58	DSa	SING	DIR		86	SDa,d	DOUB	DIR
	59	DDa	DOUB	DIR		C6	SQa,d	QUAD	DIR
	5A	DQa	QUAD	DIR		27	SS,d	SING	IND
	18	DS	SING	IND		47	SD,d	DOUB	IND
	19	DD	DOUB	IND	87	SQ,d	QUAD	IND	
	1A	DQ	QUAD	IND	<b>NOP</b>	00	no ASCII		
<b>EO</b> EX OR	6C	EOSa,d	SING	DIR		03	no ASCII		
	8C	EODa,d	DOUB	DIR		05	no ASCII		
	CC	EOQa,d	QUAD	DIR		07	no ASCII		
	2D	EOS,d	SING	IND		09	no ASCII		
	4D	EOD,d	DOUB	IND		0B	no ASCII		
	8D	EOQ,d	QUAD	IND		0D	no ASCII		
						FF	no ASCII		
<b>I</b> INCR	54	ISa	SING	DIR	3C	0-255			
	55	IDa	DOUB	DIR	3D	256-511			
	56	IQa	QUAD	DIR	3E	512-767			
	14	IS	SING	IND	3F	768-1023			
	15	ID	DOUB	IND	<b>T</b> TIER	5C	Tt,x (channels 0-255)		
	16	IQ	QUAD	IND		5D	Tt,x (channels 256-511)		
<b>x</b> CHAN ID						5E	Tt,x (channels 512-767)		
						5F	Tt,x (channels 768-1023)		

# APPENDIX B

## ASCII Conversion Table

HEX	BINARY	ASCII	HEX	BINARY	ASCII	HEX	BINARY	ASCII
00	0000 0000	NUL	2B	0010 1011	+	55	0101 0101	U
01	0000 0001	SOH	2C	0010 1100	,	56	0101 0110	V
02	0000 0010	STX	2D	0010 1101	-	57	0101 0111	W
03	0000 0011	ETX	2E	0010 1110	.	58	0101 1000	X
04	0000 0100	EOT	2F	0010 1111	/	59	0101 1001	Y
05	0000 0101	ENQ				5A	0101 1010	Z
06	0000 0110	ACK	30	0011 0000	0	5B	0101 1011	[
07	0000 0111	BEL	31	0011 0001	1	5C	0101 1100	\
08	0000 1000	BS	32	0011 0010	2	5D	0101 1101	]
09	0000 1001	TAB	33	0011 0011	3	5E	0101 1110	^
0A	0000 1010	LF	34	0011 0100	4	5F	0101 1111	_
0B	0000 1011	VT	35	0011 0101	5			
0C	0000 1100	FF	36	0011 0110	6	60	0110 0000	\
0D	0000 1101	CR	37	0011 0111	7	61	0110 0001	a
0E	0000 1110	S0	38	0011 1000	8	62	0110 0010	b
0F	0000 1111	S1	39	0011 1001	9	63	0110 0011	c
			3A	0011 1010	:	64	0110 0100	d
10	0001 0000	DEL	3B	0011 1011	;	65	0110 0101	e
11	0001 0001	DC1	3C	0011 1100	<	66	0110 0110	f
12	0001 0010	DC2	3D	0011 1101	=	67	0110 0111	g
13	0001 0011	DC3	3E	0011 1110	>	68	0110 1000	h
14	0001 0100	DC4	3F	0011 1111	?	69	0110 1001	i
15	0001 0101	NAK				6A	0110 1010	j
16	0001 0110	SYN	40	0100 0000	@	6B	0110 1011	k
17	0001 0111	ETB	41	0100 0001	A	6C	0110 1100	l
18	0001 1000	CAN	42	0100 0010	B	6D	0110 1101	m
19	0001 1001	EM	43	0100 0011	C	6E	0110 1110	n
1A	0001 1010	SUB	44	0100 0100	D	6F	0110 1111	o
1B	0001 1011	ESC	45	0100 0101	E			
1C	0001 1100	FS	46	0100 0110	F	70	0111 0000	p
1D	0001 1101	GS	47	0100 0111	G	71	0111 0001	q
1E	0001 1110	RS	48	0100 1000	H	72	0111 0010	r
1F	0001 1111	US	49	0100 1001	I	73	0111 0011	s
			4A	0100 1010	J	74	0111 0100	t
20	0010 0000	SP	4B	0100 1011	K	75	0111 0101	u
21	0010 0001	!	4C	0100 1100	L	76	0111 0110	v
22	0010 0010	"	4D	0100 1101	M	77	0111 0111	w
23	0010 0011	#	4E	0100 1110	N	78	0111 1000	x
24	0010 0100	\$	4F	0100 1111	O	79	0111 1001	y
25	0010 0101	%				7A	0111 1010	z
26	0010 0110	&	50	0101 0000	P	7B	0111 1011	{
27	0010 0111	'	51	0101 0001	Q	7C	0111 1100	
28	0010 1000	(	52	0101 0010	R	7D	0111 1101	}
29	0010 1001	)	53	0101 0011	S	7E	0111 1110	~
2A	0010 1010	*	54	0101 0100	T	7F	0111 1111	DEL





[www.peaktronics.com](http://www.peaktronics.com)

**PEAKTRONICS, Inc.** 1363 Anderson Clawson, MI 48017 Phone (248) 542-5640 FAX (248) 542-5643