Cutler-Hammer ELC Serial Driver Help

© 2012 Kepware Technologies

Table of Contents

Table of Contents ... 2
Cutler-Hammer ELC Serial Driver Help ... 3
Overview ... 3
Device Setup ... 4
Modem Setup ... 5
Cable Diagram ... 5
Data Types Description ... 7
Address Descriptions ... 8
Error Descriptions ... 11
Address Validation ... 11
Missing address ... 11
Device address '<address>' contains a syntax error ... 11
Address '<address>' is out of range for the specified device or register ... 12
Device address '<address>' is not supported by model '<model name>' ... 12
Data Type '<type>' is not valid for device address '<address>' ... 12
Device address '<address>' is Read Only ... 12
Array size is out of range for address '<address>' ... 12
Array support is not available for the specified address: '<address>' ... 13
Serial Communications ... 13
COMn does not exist ... 13
Error opening COMn ... 13
COMn is in use by another application ... 13
Unable to set comm parameters on COMn ... 13
Communications error on '<channel name>' [<error mask>] ... 14
Device Status Messages ... 14
Device '<device name>' is not responding ... 14
Unable to write to '<address>' on device '<device name>' ... 14
Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>' ... 15
Device Specific Messages ... 15
Bad address in block [<start address> to <end address>] on device '<device name>' ... 16
Bad array spanning ['<address>' to '<address>'] on device '<device name>' ... 16
Index ... 17

Cutler-Hammer ELC Serial Driver Help

Help version 1.037

CONTENTS

Overview
What is the Cutler-Hammer ELC Serial Driver?

Device Setup
How do I configure a device for use with this driver?

Data Types Description
What data types does this driver support?

Address Descriptions
How do I address a data location on a Cutler-Hammer ELC Serial device?

Error Descriptions
What error messages does the Cutler-Hammer ELC Serial device produce?

Overview

The Cutler-Hammer ELC Serial Driver provides an easy and reliable way to connect Cutler-Hammer Eaton Logic Controllers (ELC) to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. This driver is intended for use with Cutler-Hammer ELC Serial devices.

Device Setup

Supported Devices

The following Cutler-Hammer ELC models are supported:

PA10 Series
PB14 Series
PC12 Series
PV28 Series

Protocol Selection

Options include Modbus ASCII Protocol and Modbus RTU. By default, the ELC assumes communication will be in Modbus ASCII. For information on toggling from ASCII to RTU, refer to **Enabling Modbus RTU**.

Supported Communication Parameters

Baud Rate: 9600, 19200, 38400, 57600, 115200
Parity: Odd, Even, None
Data Bits: 7, 8
Stop Bits: 1,2

Note: Some devices may not support the listed configurations.

Maximum Number of Channels and Devices

The maximum number of channels supported by this driver is 100; the maximum number of devices is 32 per channel.

Ethernet Encapsulation

This driver supports Ethernet Encapsulation. Ethernet Encapsulation allows the driver to communicate with serial devices attached to an Ethernet network using a terminal server. Ethernet Encapsulation mode is invoked by selecting it from the COM ID dialog in Channel Properties. For more information on Ethernet Encapsulation, refer to the main OPC Server help file.

Device ID (PLC Network Address)

Devices are assigned Device IDs in the range 0 to 255.

Data Encoding

First Word Low: Two consecutive 16 bit registers' addresses in an ELC device are used for 32 bit data types. Users can specify whether the driver should assume the first word is the low or the high word of the 32 bit value. Check this box to use the same word order as the ELCSoft programming software.

Flow Control

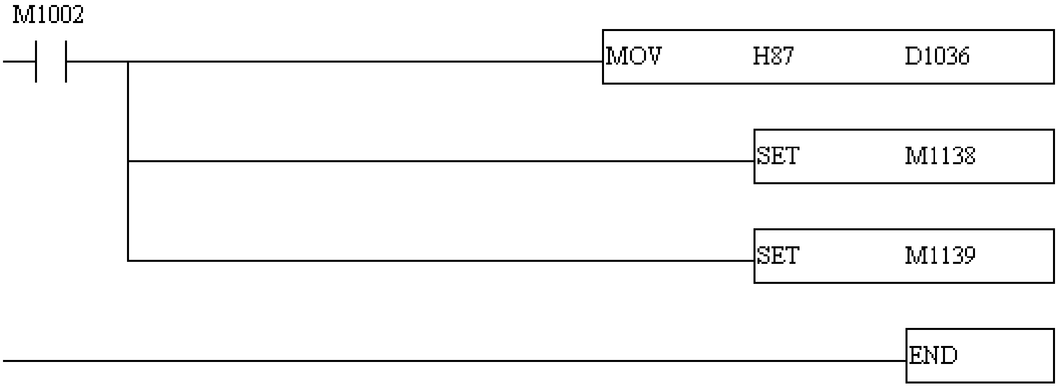
When using an RS232/RS485 converter, the type of flow control that is required depends on the needs of the converter. Some converters do not require any flow control and others will require RTS flow. Consult the converter's documentation in order to determine its flow requirements. We recommend using an RS485 converter that provides automatic flow control.

Note: When using the manufacturer's supplied communications cable, it is sometimes necessary to choose a flow control setting of **RTS** or **RTS Always** under the Channel Properties.

The Cutler-Hammer ELC Serial Serial driver supports the RTS Manual flow control option. This selection is used to configure the driver for operation with radio modems that require special RTS timing characteristics. For more information on RTS Manual flow control, refer to the main OPC Server help file topic Channel Wizard.

Enabling Modbus RTU

Whenever the ELC is rebooted, it assumes the communication protocol is Modbus ASCII. To use Modbus RTU, an RTU switching ladder logic needs to be created to toggle from ASCII to RTU.



Ladder Logic Notes

- M1002 is normally open and is only pulsed closed once, at the first scan through the logic after the ELC boots up. This drives the rest of the logic.
- MOV H87 D1036 moves the hex value 0x87 into word register D1036, which the PLC examines whenever RTU/ASCII modes are toggled.
- SET M1138 sets the bit memory m1138 ON. M1138 is referred to as "COM1 (RS-232) communication protocol settings" in the ELC help.
- SET 1139 is similar to SET M1138. M1139 is referred to in the ELC memory as "COM1 (RS-232) ASCII/RTU mode when ELC is slave. Off = ASCII, On = RT.

Block Sizes

Discrete Output
The output block size for ELC memory types S, Y, To, M and Co may be between 8 and 1024 in multiples of 8. The default setting is 256.

Discrete Input
The input block size for ELC memory type X may be between 8 and 1024 in multiples of 8. The default setting is 256.

Holding Registers
The holding register size for ELC memory types C, T and D may be between 1 and 120. The default setting is 12. The number of 32 bit counter values that can be blocked together will be one half the number of the holding registers setting, thus yielding an equivalent number of bytes per block.

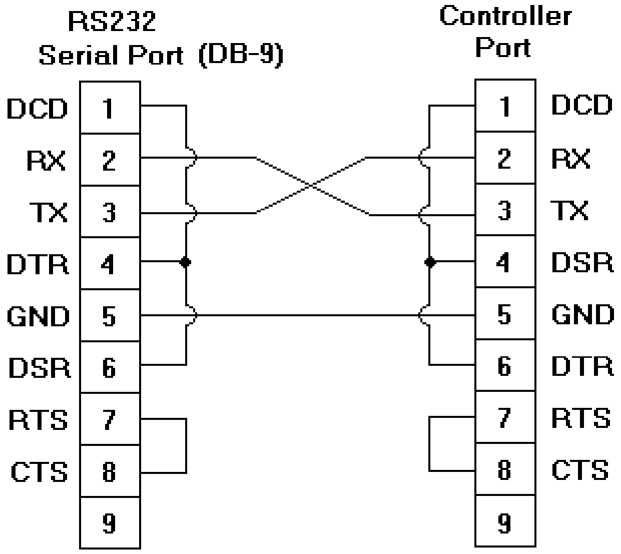
Note: Default values are best for PA10, PB14 and PC12 models. PV28 Series devices are capable of handling larger requests, however, and performance gains may be achieved by increasing the block sizes.

Modem Setup

This driver supports modem functionality. For more information, please refer to the topic "Modem Support" in the OPC Server Help documentation.

Cable Diagram

Cable Connections


RS232
Serial Port (DB-9)
Controller
Port
DCD 1
RX 2
TX 3
DTR 4
GND 5
DSR 6
RTS 7
CTS 8
9
1 DCD
2 RX
3 TX
4 DSR
5 GND
6 DTR
7 RTS
8 CTS
9

Data Types Description

Data Type	Description
Boolean	Single bit
Word	Unsigned 16 bit value bit 0 is the low bit bit 15 is the high bit
Short	Signed 16 bit value bit 0 is the low bit bit 14 is the high bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the low bit bit 31 is the high bit
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.
String	Null terminated ASCII string Supported on all models and includes HiLo LoHi byte order selection.
Double*	64 bit Floating point value The driver interprets four consecutive registers as a double precision value by making the last two registers the high DWord and the first two registers the low DWord.
Double Example	If register 40001 is specified as a double, bit 0 of register 40001 would be bit 0 of the 64 bit data type and bit 15 of register 40004 would be bit 63 of the 64 bit data type.
Float*	32 bit Floating point value The driver interprets two consecutive registers as a single precision value by making the last register the high word and the first register the low word.
Float Example	If register 40001 is specified as a Float, bit 0 of register 40001 would be bit 0 of the 32 bit data type and bit 15 of register 40002 would be bit 31 of the 32 bit data type.

*The descriptions assume the default first DWord low data handling of 64 bit data types, and first word low data handling of 32 bit data types.

Address Descriptions

Addressing information is shown in tables below for the **PA10/PC12 Series**, **PB14 Series** and **PV28 Series**. The default data types for dynamically defined tags are shown in **bold** where appropriate.

Note: For notes and restrictions, refer to **Write Only Access**, **String Support** and **Array Support**.

PA10 Series/PC12 Series

Memory Type	Range	Data Type	Access*	Remarks
Discrete Input (external I/O)	X0-X177 (octal)	**Boolean**	Read Only	
Discrete Output (external I/O)	Y0-Y177 (octal)	**Boolean**	Read/Write	
Main Relay	M0-M4095	**Boolean**	Read/Write	
Timer Status	To0-To255	**Boolean**	Read/Write	Contact = ON when timer reaches preset value
Counter Status (16 bit counters)	Co0-Co199	**Boolean**	Read/Write	Contact = ON when counter reaches preset value
Counter Status (32 bit counters)	Co200-Co255	**Boolean**	Read/Write	Contact = ON when counter reaches preset value
Step Point	S0-S1023	**Boolean**	Read/Write	Sequential Function Chart usage
Timer Current Value	T0-T255 Txxx.0-Txxx.15	**Word**, Short, BCD Float, Long, LBCD Double, DWord Boolean	Read/Write	Represents the current value of the 1mS, 10mS, or 100mS timer
Counter Current Value (16 bit counters)	C0-C199 Cxxx.0-Cxxx.15	**Word**, Short, BCD Float, Long, LBCD Double, DWord Boolean	Read/Write	Represents the current value of the counter
Counter Current Value (32 bit counters)	C200-C255 Cxxx.0-Cxxx.31	**DWord**, Float Long, LBCD, Double Boolean	Read/Write Read Only	Represents the current value of the counter
Data Register	D0-D4999 Dxxx.0-Dxxx.15	**Word**, Short, BCD Float, Long, LBCD Double, DWord Boolean	Read/Write	General data storage
Data Register as String with HiLo Byte Order	D0.2H-D4984.32H	**String****	Read/Write	.Bit is string length, range 2 to 32 bytes
Data Register as String with LoHi Byte Order	D0.2L-D4984.32L	**String****	Read/Write	.Bit is string length, range 2 to 32 bytes

PB14 Series

Memory Type	Range	Data Type	Access*	Remarks
Discrete Input (external I/O)	X0-X177 (octal)	**Boolean**	Read Only	
Discrete Output (external I/O)	Y0-Y177 (octal)	**Boolean**	Read/Write	
Main Relay	M0-M1279	**Boolean**	Read/Write	
Main Relay	M2000-M4095	**Boolean**	Read/Write	
Timer Status	To0-To127	**Boolean**	Read/Write	Contact = ON when timer reaches preset value
Counter Status (16 bit counters)	Co0-Co127	**Boolean**	Read/Write	Contact = ON when counter reaches preset value
Counter Status (32 bit counters)	Co232-Co255	**Boolean**	Read/Write	Contact = ON when counter reaches

				preset value
Step Point	S0-S127	**Boolean**	Read/Write	Sequential Function Chart usage
Timer Current Value	T0-T127 Txxx.0-Txxx.15	**Word**, Short, BCD Float, Long, LBCD Double, DWord Boolean	Read/Write	Represents the current value of the 1mS, 10mS or 100mS timer
Counter Current Value (16 bit counters)	C0-C127 Cxxx.0-Cxxx.15	**Word**, Short, BCD Float, Long, LBCD Double, DWord Boolean	Read/Write	Represents the current value of the counter
Counter Current Value (32 bit counters)	C232-C255 Cxxx.0-Cxxx.31	**DWord**, Float Long, LBCD, Double Boolean	Read/Write Read Only	Represents the current value of the counter
Data Register	D0-D599 Dxxx.0-Dxxx.15	**Word**, Short, BCD Float, Long, LBCD Double, DWord Boolean	Read/Write	General data storage
Data Register	D1000-D1311 Dxxx.0-Dxxx.15	**Word**, Short, BCD Float, Long, LBCD Double, DWord Boolean	Read/Write	General data storage
Data Register as String with HiLo Byte Order	D0.2H-D584.32H	**String****	Read/Write	.Bit is string length, range 2 to 32 bytes
Data Register as String with LoHi Byte Order	D0.2L-D584.32L	**String****	Read/Write	.Bit is string length, range 2 to 32 bytes
Data Register as String with HiLo Byte Order	D1000.2H-D1296.32H	**String****	Read/Write	.Bit is string length, range 2 to 32 bytes
Data Register as String with LoHi Byte Order	D1000.2L-D1296.32L	**String****	Read/Write	.Bit is string length, range 2 to 32 bytes

PV28 Series

Memory Type	**Range**	**Data Type**	**Access***	**Remarks**
Discrete Input (external I/O)	X0-X377 (octal)	**Boolean**	Read Only	
Discrete Output (external I/O)	Y0-Y377 (octal)	**Boolean**	Read/Write	
Main Relay	M0-M4095	**Boolean**	Read/Write	
Timer Status	To0-To255	**Boolean**	Read/Write	Contact = ON when timer reaches preset value
Counter Status (16 bit counters)	Co0-Co199	**Boolean**	Read/Write	Contact = ON when counter reaches preset value
Counter Status (32 bit counters)	Co200-Co255	**Boolean**	Read/Write	Contact = ON when counter reaches preset value
Step Point	S0-S1023	**Boolean**	Read/Write	Sequential Function Chart usage
Timer Current Value	T0-T255 Txxx.0-Txxx.15	**Word,** Short, BCD Float, DWord, Long LBCD, Double Boolean	Read/Write	Represents the current value of the 1mS, 10mS, or 100mS timers
Counter Current Value (16 bit counters)	C0-C199 Cxxx.0-Cxxx.15	**Word**, Short, BCD Float, DWord, Long LBCD, Double Boolean	Read/Write	Represents the current value of the counter

Counter Current Value (32 bit counters)	C200-C255 Cxxx.0-Cxxx.31	**DWord**, Float Long, LBCD, Double Boolean	Read/Write Read Only	Represents the current value of the counter
Data Register	D0-D9999 Dxxx.0-Dxxx.15	**Word**, Short, BCD Float, DWord, Long LBCD, Double Boolean	Read/Write	General data storage
Data Register as String with HiLo Byte Order	D0.2H-D9984.32H	**String****	Read/Write	.Bit is string length, range 2 to 32 bytes
Data Register as String with LoHi Byte Order	D0.2L-D9984.32L	**String****	Read/Write	.Bit is string length, range 2 to 32 bytes

*Write Only Access

All Read/Write addresses may be set as Write by prefixing a "W" to the address such as "WD0", which will prevent the driver from reading the register at the specified address. Any attempts by the client to read a write-only tag will result in obtaining the last successful write value to the specified address. If no successful writes have occurred, then the client will receive 0/NULL for numeric/string values for an initial value.

Caution: Setting the "Client access" privileges of write-only tags to Read Only will cause writes to these tags to fail and the client to always receive 0/NULL for numeric/string values.

**String Support

The ELC devices support reading and writing data register memory as an ASCII string. When using data registers for string data, each register will contain two bytes of ASCII data. The order of the ASCII data within a given register can be selected when the string is defined. The length of the string can be from 2 to 32 bytes and is entered in place of a bit number. The length must be entered as an even number. The byte order is specified by appending either an "H" or "L" to the address.

Examples
1. To address a string starting at D200 with a length of 32 bytes and HiLo byte order, enter:
D200.32H

2. To address a string starting at D500 with a length of 32 bytes and LoHi byte order, enter:
D500.32L

Note: The string length may be limited by the maximum size of the write request that your device will allow. If while utilizing a string tag you receive an error message in the server event window of "Unable to write to address <address> on device <device>: Device responded with exception code 3." This means the device did not accept the length of your string. If possible, try shortening the string.

Normal Address Example

The 177'th (octal) output coil would be addressed as 'Y177' using octal addressing.

Array Support

Arrays are supported for registers (i.e. T, C, and D addresses) except for bit-within-word and strings. Arrays are also supported for input and output coils (Boolean data types, i.e. M, S, X, Y, To, and Co addresses). There are two methods of addressing an array. Examples are given using register addresses.

Dxxx [rows] [cols]
Dxxx [cols] this method assumes rows is equal to one

For arrays, rows multiplied by cols cannot exceed the block size that has been assigned to the device for the register/coil type. For register arrays of 32 bit data types, rows multiplied by cols multiplied by 2 cannot exceed the block size.

Error Descriptions

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation

Missing address
Device address '<address>' contains a syntax error
Address '<address>' is out of range for the specified device or register
Device address '<address>' is not supported by model '<model name>'
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only
Array size is out of range for address '<address>'
Array support is not available for the specified address: '<address>'

Serial Communications

COMn does not exist
Error opening COMn
COMn is in use by another application
Unable to set comm parameters on COMn
Communications error on '<channel name>' [<error mask>]

Device Status Messages

Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'
Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'

Device Specific Messages

Bad address in block [<start address> to <end address>] on device '<device name>'
Bad array spanning ['<address>' to '<address>'] on device '<device name>'

Address Validation

The following error/warning messages may be generated. Click on the link for a description of the message.

Address Validation

Missing address
Device address '<address>' contains a syntax error
Address '<address>' is out of range for the specified device or register
Device address '<address>' is not supported by model '<model name>'
Data Type '<type>' is not valid for device address '<address>'
Device address '<address>' is Read Only
Array size is out of range for address '<address>'
Array support is not available for the specified address: '<address>'

Missing address

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has no length.

Solution:
Re-enter the address in the client application.

Device address '<address>' contains a syntax error

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically contains one or more invalid characters.

Solution:
Re-enter the address in the client application.

Address '<address>' is out of range for the specified device or register

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically references a location that is beyond the range of supported locations for the device.

Solution:
Verify the address is correct; if it is not, re-enter it in the client application.

Device address '<address>' is not supported by model '<model name>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically references a location that is valid for the communications protocol but not supported by the target device.

Solution:
Verify that the address is correct; if it is not, re-enter it in the client application. Also, verify that the selected model name for the device is correct.

Data Type '<type>' is not valid for device address '<address>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has been assigned an invalid data type.

Solution:
Modify the requested data type in the client application.

Device address '<address>' is Read Only

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically has a requested access mode that is not compatible with what the device supports for that address.

Solution:
Change the access mode in the client application.

Array size is out of range for address '<address>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically is requesting an array size that is too large for the address type or block size of the driver.

Solution:
Re-enter the address in the client application to specify a smaller value for the array or a different starting point.

Array support is not available for the specified address: '<address>'

Error Type:
Warning

Possible Cause:
A tag address that has been specified statically contains an array reference for an address type that doesn't support arrays.

Solution:
Re-enter the address in the client application to remove the array reference or correct the address type.

Serial Communications

The following error/warning messages may be generated. Click on the link for a description of the message.

Serial Communications
COMn does not exist
Error opening COMn
COMn is in use by another application
Unable to set comm parameters on COMn
Communications error on '<channel name>' [<error mask>]

COMn does not exist

Error Type:
Fatal

Possible Cause:
The specified COM port is not present on the target computer.

Solution:
Verify that the proper COM port has been selected.

Error opening COMn

Error Type:
Fatal

Possible Cause:
The specified COM port could not be opened due an internal hardware or software problem on the target computer.

Solution:
Verify that the COM port is functional and may be accessed by other Windows applications.

COMn is in use by another application

Error Type:
Fatal

Possible Cause:
The serial port assigned to a device is being used by another application.

Solution:
Verify that the correct port has been assigned to the channel and that only one copy of the current project is running.

Unable to set comm parameters on COMn

Error Type:
Fatal

Possible Cause:
The serial parameters for the specified COM port are not valid.

Solution:
Verify the serial parameters and make any necessary changes.

Communications error on '<channel name>' [<error mask>]

Error Type:
Serious

Error Mask Definitions:
B = Hardware break detected.
F = Framing error.
E = I/O error.
O = Character buffer overrun.
R = RX buffer overrun.
P = Received byte parity error.
T = TX buffer full.

Possible Cause:
Either the serial connection between the device and the Host PC is bad or the communications parameters for the serial connection are incorrect.

Solution:
Verify the cabling between the PC and the PLC device. Also, verify that the specified communications parameters match those of the device.

Device Status Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Device Status Messages
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'
Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'

Device '<device name>' is not responding

Error Type:
Serious

Possible Cause:
1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the Request Timeout setting so that the entire response can be handled.

Unable to write to '<address>' on device '<device name>'

Error Type:
Serious

Possible Cause:
1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:
1. Verify the cabling between the PC and the PLC device.

2. Verify that the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.

Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>'

Error Type:
Warning

Possible Cause:
Please refer to the Exception Codes table below.

Solution:
Solution will depend upon the Exception Code. Please refer to table below.

Modbus Exception Codes

The following information is from the Modbus Application Protocol Specifications documentation.

Code Dec/Hex	Name	Meaning
01/0x01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the server (or slave). This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the server (or slave) is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02/0x02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the server (or slave). More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed, a request with offset 96 and length 5 will generate exception 02.
03/0x03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for server (or slave). This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does NOT mean that a data item submitted for storage in a register has a value outside the expectation of the application program, since the MODBUS protocol is unaware of the significance of any particular value of any particular register.
04/0x04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.
05/0x05	ACKNOWLEDGE	The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the master. The master can next issue a Poll Program Complete message to determine if processing is completed.
06/0x06	SLAVE DEVICE BUSY	The slave is engaged in processing a long-duration program command. The master should retransmit the message later when the slave is free.
07/0x07	NEGATIVE ACKNOWLEDGE	The slave cannot perform the program function received in the query. This code is returned for an unsuccessful programming request using function code 13 or 14 decimal. The master should request diagnostic or error information from the slave.
08/0x08	MEMORY PARITY ERROR	The slave attempted to read extended memory, but detected a parity error in the memory. The master can retry the request, but service may be required on the slave device.
10/0x0A	GATEWAY PATH UNAVAILABLE	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. This usually means that the gateway is misconfigured or overloaded.
11/0x0B	GATEWAY TARGET DEVICE FAILED TO RESPOND	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

Device Specific Messages

The following error/warning messages may be generated. Click on the link for a description of the message.

Device Specific Messages

Bad address in block [<start address> to <end address>] on device '<device name>'
Bad array spanning ['<address>' to '<address>'] on device '<device name>'

Bad address in block [<start address> to <end address>] on device '<device name>'

Error Type:
Serious

Possible Cause:
An attempt has been made to reference a nonexistent location in the specified device.

Solution:
Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations.

Bad array spanning ['<address>' to '<address>'] on device '<device name>'

Error Type:
Fatal

Possible Cause:
An array of addresses was defined that spans past the end of the address space.

Solution:
Verify the size of the device's memory space and then redefine the array length accordingly.

Index

A

Address '<address>' is out of range for the specified device or register 12

Address Descriptions 8

Address Validation 11

Array size is out of range for address '<address>' 12

Array support is not available for the specified address:'<address>' 13

ASCII/RTU Toggle 4

B

Bad address in block [<start address> to <end address>] on device '<device name>' 16

Bad array spanning ['<address>' to '<address>'] on device '<device name>' 16

BCD 7

Block Size 5

Boolean 7

C

Cable Diagram 5

Communications error on '<channel name>' [<error mask>] 14

COMn does not exist 13

COMn is in use by another application 13

D

Data Type '<type>' is not valid for device address '<address>' 12

Data Types Description 7

Device '<device name>' is not responding 14

Device address '<address>' contains a syntax error 11

Device address '<address>' is not supported by model '<model name>' 12

Device address '<address>' is Read Only 12

Device ID 4

Device Specific Messages 15

Device Status Messages 14
DWord 7

E

Error Descriptions 11
Error opening COMn 13

F

Float 7
Framing 14

H

Help Contents 3

L

LBCD 7
Long 7

M

Missing address 11
Modem Setup 5

N

Network 4

O

Overrun 14
Overview 3

P

Parity 14

R

RTU Communication 4

S

Serial Communications 13
Short 7
String 7

U

Unable to set comm parameters on COMn 13
Unable to write to '<address>' on device '<device name>' 14
Unable to write to address '<address>' on device '<device>': Device responded with exception code '<code>' 15

W

Word 7