Cutler-Hammer D50/300 Driver Help

© 2012 Kepware Technologies

Table of Contents

Cutler-Hammer D50/300 Driver Help

Help version 1.021

CONTENTS

Overview
What is the Cutler-Hammer D50/300 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 D50/300 device?

Error Descriptions
What error messages does the Cutler-Hammer D50/300 driver produce?

Overview

The Cutler-Hammer D50/300 Driver provides an easy and reliable way to connect Cutler-Hammer D50/300 devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It supports Cutler-Hammer D50 and D300 devices in addition to Allen Bradley OEMax devices.

Device Setup

Supported Devices
D50, D300, and OEMax Programmable Controllers

Communication Protocol
Cutler-Hammer Hex Mode Two Step and Four Step Protocol

Supported Communication Parameters
Baud Rate: 1200, 2400, 9600, 19200, or 38400
Parity: None, Even, or Odd
Data Bits: 5, 6, 7 or 8
Stop Bits: 1 or 2

Note: Not all devices support every listed configuration.

Ethernet Encapsulation
This driver supports Ethernet Encapsulation, which allows communications with serial devices attached to an Ethernet network using a terminal server or device server. It can be invoked through the COM ID dialog in Channel Properties. For more information, refer to the server help documentation.

Device IDs
Up to 32 devices may be defined. Valid device IDs range from 1 to 255.

Flow Control
When using an RS-232/RS-485 converter, the type of flow control that is required depends on the converter's needs. Some converters do not require any flow control, whereas others require RTS flow. To determine flow requirements, refer to the converter's documentation. An RS485 converter that provides automatic flow control is recommended.

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.

When communicating with a Cutler-Hammer D50 or D300, it is recommended that an RS-485 converter that supports automatic flow control be used. This converter should automatically switch between transmit and receive mode when its control jumper is set to SD mode.

Cable Connections
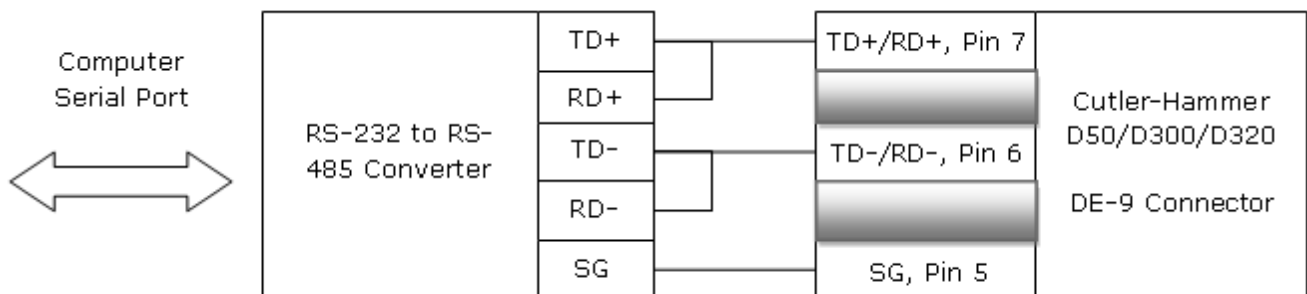
See Also: **Settings**

Settings
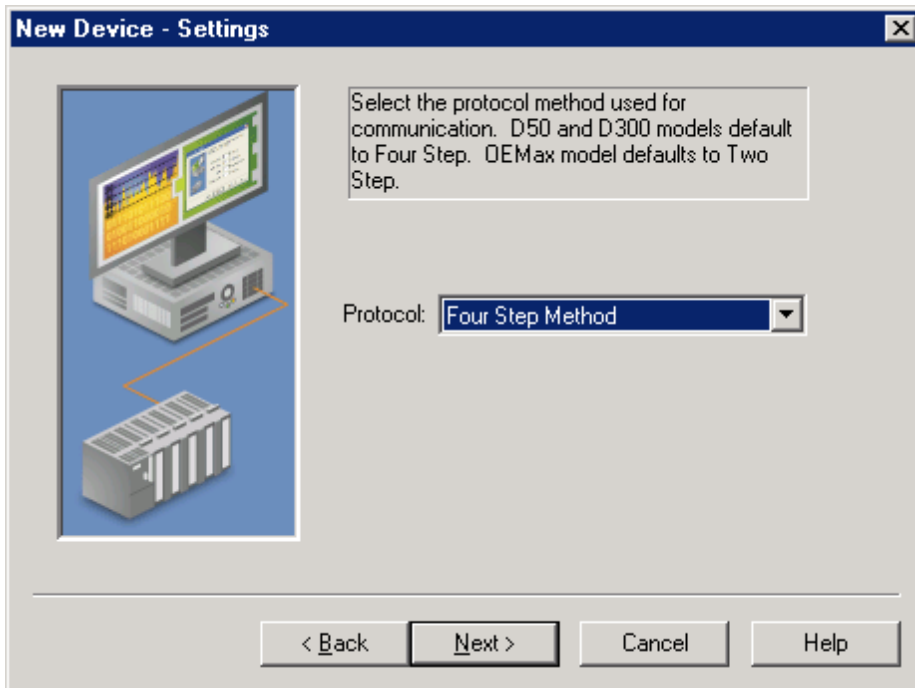
Description of the parameter is as follows:

- **Protocol:** This parameter specifies the device's communication methods. Options include Two Step Method and Four Step Method. OEMax devices default to Two Step Method. D50 and D300 devices default to Four Step Method.

Note: Not all devices support all communication methods.

Modem Setup

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

Data Types Description

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8 bit value bit 0 is the least significant bit bit 7 is the most significant bit
Char	Signed 8 bit value bit 0 is the least significant bit bit 6 is the most significant bit bit 7 is the sign bit
Word	Unsigned 16 bit value bit 0 is the least significant bit bit 15 is the most significant bit
Short	Signed 16 bit value bit 0 is the least significant bit bit 14 is the most significant bit bit 15 is the sign bit
DWord	Unsigned 32 bit value bit 0 is the least significant bit bit 31 is the most significant bit
Long	Signed 32 bit value bit 0 is the least significant bit bit 30 is the most significant bit bit 31 is the sign bit

Address Descriptions

Address specifications vary depending on the model in use. Select a link from the following list to obtain specific address information for the model of interest.

D50
D300
OEMax

D50 Addressing

The default data types are shown in **bold**.

Memory Type	Reference	Data Type
I/O	M, K, F	**Boolean**,* Char, Byte, **Word**,* Short, DWord, Long
External I/O	R	**Boolean**,* Char, Byte, **Word**,* Short, DWord, Long
Timers/Counters	TC PV, SV	**Boolean** **Word**, Short
Status	SR	Char, Byte, **Word**, Short, DWord, Long
Data Words	W	Char, Byte, **Word**, Short, DWord, Long

*A dot (.) notation is used to determine default type.

Note: The data types Char and Byte access the least significant byte of word for read and write operations. On a read operation, the least significant byte of word is returned. On a write operation, data is placed into the least significant byte and the most significant byte is set to 0.

Address Specifications

Memory Type	Discrete Memory Reference	Discrete Memory Range	Word Memory Reference	Word Memory Range
External I/O	R<xxx>.<yy> xxx - Word Number yy-Bit Number***	R000.00-R029.15	R<xxxx> xxxx-Word Number**	R0000-R0029
Internal Relay	M<xxx>.<yy> xxx-Word Number yy-Bit Number	M000.00-M031.15	M<xxxx> xxxx-Word Number**	M0000-M0031
Keep Relay	K<xxx>.<yy> xxx-Word Number yy-Bit Number	K000.00-K015.15	K<xxxx> xxxx-Word Number**	K0000-K0015
Special Flag	F<xxx>.<yy> xxx-Word Number yy-Bit Number	F000.00-F001.15	F<xxxx> xxxx-Word Number**	F0000-F0001
Timers/Counters	TC<xxx> xxx-Bit Number	TC000-TC255	N/A	
Timers/ Counters Present Values	N/A		PV<xxxx> xxxx-Word Number	PV0-PV255
Timers/Counters Set Values	N/A		SV<xxxx> xxxx-Word Number	SV0-SV255
Status Register	N/A		SR<xxx> xxxx-Word Number**	SR0-SR255
Word Register	N/A		W<xxxx> xxxx-Word Number**	W0000-W0255

**Longs and DWords are allowed from 0 to the highest word number minus 1. For example, an internal relay can be both read and written for a range of M0000-M0126.

***yy bits are only valid between 0 and 15.

Examples

W76	Word register 76
K13.12	bit 12 of keep relay word 13

D300 Addressing

The default data types are shown in **bold**.

Memory Type	Reference	Data Type
External I/O	R	**Boolean**,* Char, Byte, **Word**,* Short, DWord, Long
Timers/Counters	TC PV, SV	**Boolean** **Word**, Short
Status	SR	Char, Byte, **Word**, Short, DWord, Long
Data Words	W	Char, Byte, **Word**, Short, DWord, Long

*A dot (.) notation is used to determine default type.

Note: The data types Char and Byte access the least significant byte of word for read and write operations. On a read operation, the least significant byte of word is returned. On a write operation, data is placed into the least significant byte and the most significant byte is set to 0.

Address Specifications

Memory Type	Discrete Memory Reference	Discrete Memory Range	Word Memory Reference	Word Memory Range
External I/O	R<xxx>.<yy> xxx-Word Number yy-Bit Number***	R000.00-R033.15 R064.00-R127.15	R<xxxx> xxxx-Word Number**	R0000-R0033 R0064-R0127
Internal Relay	M<xxx>.<yy> xxx-Word Number yy-Bit Number	M000.00-M127.15	M<xxxx> xxxx-Word Number**	M0000-M0127
Keep Relay	K<xxx>.<yy> xxx-Word Number yy-Bit Number	K000.00-K127.15	K<xxxx> xxxx-Word Number**	K0000-K0127
Link Relay	L<xxx>.<yy> xxx-Word Number yy-Bit Number	L000.00-L063.15	L<xxxx> xxxx-Word Number**	L0000-L0063
Special Flag	F<xxx>.<yy> xxx-Word Number yy-Bit Number	F000.00-F015.15	F<xxxx> xxxx-Word Number**	F0000-F0015
Timers/Counters	TC<xxx> xxx-Bit Number	TC000-TC255	N/A	
Timers/Counters Present Values	N/A		PV<xxxx> xxxx-Word Number	PV0-PV255
Timers/Counters Set Values	N/A		SV<xxxx> xxxx-Word Number	SV0-SV255
Status Register	N/A		SR<xxx> xxxx-Word Number**	SR0-SR511
Word Register	N/A		W<xxxx> xxxx-Word Number**	W0000-W2047

**Longs and DWords are allowed from 0 to the highest word number minus 1. For example, an internal relay can be both read and written for a range of M0000-M0126.

***yy bits are only valid between 0 and 15.

Examples

W76	Word register 76
K13.12	bit 12 of keep relay word 13

OEMax Addressing

The default data types are shown in **bold**.

Memory Type	Reference	Data Type
I/O	M, K, F, L	**Boolean**,* Char, Byte, **Word**,* Short, DWord, Long

External I/O	R	**Boolean**,* Char, Byte, **Word**,* Short, DWord, Long
Timers/Counters	TC PV, SV	**Boolean** **Word**, Short
Status	SR	Char, Byte, **Word**, Short, DWord, Long
Data Words	W	Char, Byte, **Word**, Short, DWord, Long

*A dot (.) notation is used to determine default type.

Note: The data types Char and Byte access the least significant byte of word for read and write operations. On a read operation, the least significant byte of word is returned. On a write operation, data is placed into the least significant byte and the most significant byte is set to 0.

Address Specifications

Memory Type	Discrete Memory Reference	Discrete Memory Range	Word Memory Reference	Word Memory Range
External I/O	R<xxx>.<yy> xxx-Word Number yy-Bit Number***	R000.00-R127.15	R<xxxx> xxxx-Word Number**	R0000-R0127
Internal Relay	M<xxx>.<yy> xxx-Word Number yy-Bit Number	M000.00-M127.15	M<xxxx> xxxx-Word Number**	M0000-M0127
Keep Relay	K<xxx>.<yy> xxx-Word Number yy-Bit Number	K000.00-K127.15	K<xxxx> xxxx-Word Number**	K0000-K0127
Link Relay	L<xxx>.<yy> xxx-Word Number yy-Bit Number	L000.00-L063.15	L<xxxx> xxxx-Word Number**	L0000-L0063
Special Flag	F<xxx>.<yy> xxx-Word Number yy-Bit Number	F000.00-F0127.15	F<xxxx> xxxx-Word Number**	F0000-F0127
Timers/Counters	TC<xxx> xxx-Bit Number	TC000-TC255	N/A	
Timers/Counters Present Values	N/A		PV<xxxx> xxxx-Word Number	PV0-PV255
Timers/Counters Set Values	N/A		SV<xxxx> xxxx-Word Number	SV0-SV255
Status Register	N/A		SR<xxx> xxxx-Word Number**	SR0-SR511
Word Register	N/A		W<xxxx> xxxx-Word Number**	W0000-W2047 W3072-W5119

**Longs and DWords are allowed from 0 to the highest word number minus 1. For example, an internal relay can be both read and written for a range of M0000-M0126.

***yy bits are only valid between 0 and 15.

Examples

| W76 | Word register 76 |
| K13.12 | bit 12 of keep relay word 13 |

Error Descriptions

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

Address Validation Error Messages
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

Serial Communications Error Messages
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 Error Messages
Device '<device name>' is not responding
Unable to write to '<address>' on device '<device name>'

Device Specific Error Messages
Bad address in block [<start address> to <end address>] on device '<device name>'

Address Validation Error Messages

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

Missing address

Error Type:
Warning

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

Solution:

Verify 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 dynamically 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 dynamically 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.

Serial Communications Error Messages

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.

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:
1. The serial connection between the device and the Host PC is bad.
2. The communications parameters for the serial connection are incorrect.

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify the specified communications parameters match those of the device.

Device Status Error Messages

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

Serial Communications Error Messages
Device '<device name>' not responding
Unable to write to '<address>' on device '<device name>'

Device '<device name>' 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.

Device Specific Error Messages

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

Device Specific Error Messages
Bad address in block [<start address> to <end 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.

Index