

# **Contrex M-Series Driver Help**

**© 2012 Kepware Technologies**

# Table of Contents

Table of Contents .....	2
Contrex M-Series Driver Help .....	4
Overview .....	4
Device Setup .....	5
Modem Setup .....	5
Cable Diagram .....	5
Settings .....	6
Data Types Description .....	7
Address Descriptions .....	8
M-Cut Addressing .....	8
M-Drive Addressing .....	8
M-Rotary Addressing .....	9
M-Shuttle Addressing .....	10
M-Track Addressing .....	10
M-Traverse Addressing .....	11
M-Trim Addressing .....	11
ML-Drive Addressing .....	12
ML-Trim Addressing .....	12
Error Descriptions .....	14
Address Validation .....	14
Missing address .....	14
Device address '<address>' contains a syntax error .....	14
Address '<address>' is out of range for the specified device or register .....	15
Device address '<address>' is not supported by model '<model name>' .....	15
Data Type '<type>' is not valid for device address '<address>' .....	15
Device address '<address>' is Read Only .....	15
Serial Communications .....	15
COMn does not exist .....	15
Error opening COMn .....	16
COMn is in use by another application .....	16
Unable to set comm parameters on COMn .....	16
Communications error on '<channel name>' [<error mask>] .....	16
Device Status Messages .....	16
Device '<device name>' is not responding .....	17
Device Specific Messages .....	17
Unable to write to '<address>' on device '<device name>.' Contrex Error (<error code>) <error descrip- tion> .....	17
Deactivating Tag '<address>' on device '<device name>.' Contrex Error (<error code>) <error descrip- tion> .....	17

Index ..... 19

## Contrex M-Series Driver Help

---

Help version 1.011

### CONTENTS

#### [Overview](#)

What is the Contrex M-Series 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 Contrex M-Series device?

#### [Error Descriptions](#)

What error messages does the Contrex M-Series driver produce?

### Overview

---

The Contrex M-Series Driver provides an easy and reliable way to connect Contrex M-Series devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications.

## Device Setup

---

### Supported Devices

M-Cut  
M-Drive  
M-Rotary  
M-Shuttle  
M-Track  
M-Traverse  
M-Trim  
ML-Drive  
ML-Trim

### Communication Protocol

M-Series Data-Link Protocol (4 byte/ 8 byte).  
M-Drive, M-Trim, ML-Drive and ML-Trim utilize the 4 byte protocol.  
M-Cut, M-Rotary, M-Shuttle, M-Track, and M-Traverse utilize the 8 byte protocol.

### Supported Communication Parameters\*

Baud Rate: 300, 600, 1200, 2400, 9600\*\*  
Parity: None, Even  
Data Bits: 7, 8  
Stop Bits: 1, 2

\*Not all devices support the listed configurations. Consult the device documentation and configuration for more details.

\*\*This is the default setting.

### 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, refer to the main OPC Server help file.

### Device IDs

Valid Device IDs range from 0 to 32.

Device ID 0 is used to identify broadcast messages. A device topic can be configured within a server that uses a Device ID of station 0. All tags and data associated with this broadcast topic will be treated as Write Only. By defining a broadcast topic in the Server project, an M-Series command may be issued to multiple controllers simultaneously.

**Note:** Selecting a broadcast device with a 4 byte model (M-Drive, M-Trim, ML-Drive or ML-Trim) will utilize the 4 byte protocol. Therefore, 8 byte devices (M-Cut, M-Rotary, M-Shuttle, M-Track, and M-Traverse) coexisting on the same channel will not process these messages. The reverse is true if the broadcast device is set to use an 8 byte model.

### Flow Control

When using an RS232/RS485 converter, the type of flow control that is required will depend upon the needs of the converter. Some converters do not require any flow control and others will require RTS flow. Consult the converter's documentation to determine its flow requirements.

**See Also:** [Cable Diagram](#) and [Settings](#).

## 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

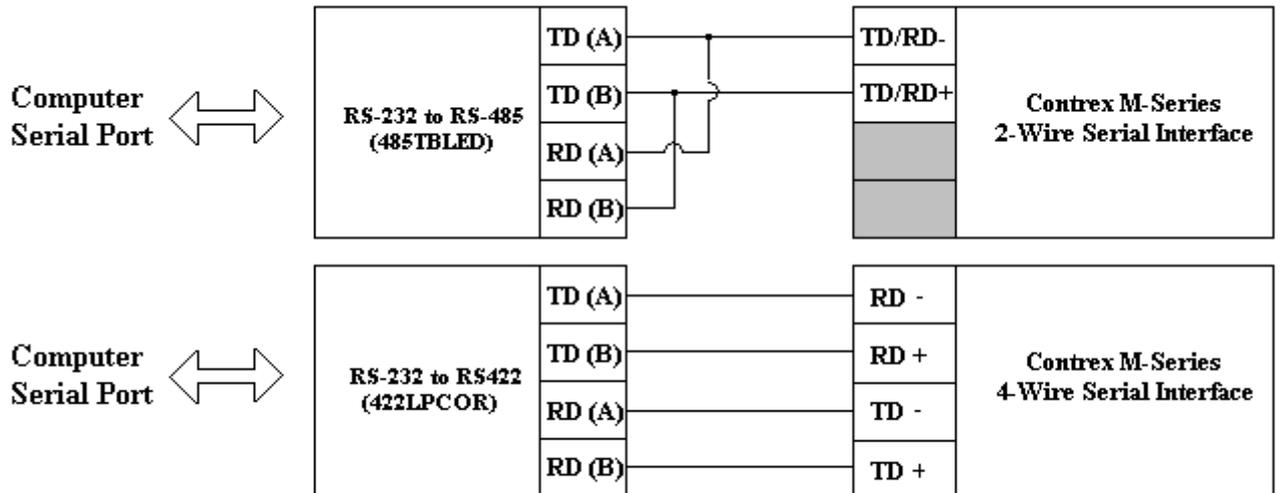
---

### Cable Connections

We suggest using a high quality manufactured R232/485 converter (485TBLED) when communicating with the Contrex M-Series devices. This converter will automatically handle the operation of the 485 line drivers. Also, for optimal performance, it was required that the 485 converter's receiver be turned on at all times due to the fast

turn around of the M-Series devices. Some converters refer to this as **Echo On**. In order for the driver to deal with echo on, set the Echo On parameter described above.

The following diagram illustrates an RS-232 to RS-485 cable diagram using this converter. The RS-232 to RS-422 converter can be used with the four-wire connection of the M-TRIM and others. When using an RS-422 connection, make sure the Echo On setting is disabled.



## Settings

### Precision

Set the default Floating point precision to use when writing Float data to the M-Series devices. This default setting can be overridden on a per tag basis by appending a dot-precision number. For example, CP20 -> append precision 1 -> CP20.1.

The driver will force all Float data written to the controllers to conform to these precision settings. For example, CP20.2 and Float data 25.5125 is sent as 25.51. It is possible that some parameter locations will reject a precision. Consult the device documentation on parameter/precision.

**Note:** For many of the M-Series devices such as the ML-Drive, only two parameters allow for precision, while all other parameters have fixed decimal notation. Set the master precision to 0. The two parameters CP20 and CP21 that allow precision can be set using the dot-precision notation.

### Echo On

If the RS232/485 converter's receiver is always on set this parameter. This selection should be disabled when using an RS-422 converter. For more information, refer to [Cable Diagram](#).

**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
DWord Example	If register 40001 is specified as a DWord, 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 reverse is true when this is not selected.
Long	Signed 32 bit value bit 0 is the low bit bit 30 is the high bit bit 31 is the sign bit
Long Example	If register 40001 is specified as a long, 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 reverse is true when this is not selected.
Float	32 bit Floating point value The driver interprets two consecutive registers as a floating-point value by making the first register the low word and the second register the high 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 reverse is true when this is not selected.

## 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.

[M-Cut](#)  
[M-Drive](#)  
[M-Rotary](#)  
[M-Shuttle](#)  
[M-Track](#)  
[M-Traverse](#)  
[M-Trim](#)  
[ML-Drive](#)  
[ML-Trim](#)

**Note:** The actual number of available addresses depends on the Contrex M-Series device in use. Refer to the device documentation for address ranges.

## M-Cut Addressing

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Long, DWord, <b>Float</b>	Read Only
Control Parameters	CP00-CP99	Long, DWord, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Stop
02	Halt
03	Batch Reset
04	Resume Move
05	Move
06	Home Return
07	Back Step
08	Home To Index
09	Home Set
10	Jog Start
11	Jog Stop
12	Forward Select
13	Reverse Select
14	Profile 1 Select
15	Profile 2 Select
16	Profile 3 Select
17	Profile 4 Select

For more information on these command control codes, consult the device documentation.

### Examples

1. To send the **Stop** command to the controller, use the syntax CC0.1 and write to the address.
2. To initiate a **Home Return**, use the syntax CC0.6 and write to the address.

## M-Drive Addressing

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Short, Word, <b>Float</b>	Read Only
Control Parameters	CP00-CP99	Short, Word, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Fast Stop
02	Ramp Stop
03	Run
04	Enable Primary Mode
05	Enable Secondary Mode
08	Enable Closed Loop
09	Enable Open Loop
10	Enable Primary Setpoint 1
11	Enable Primary Setpoint 2
12	Enable Secondary Setpoint 3
13	Enable Secondary Setpoint 4
14	Ramp Disable
15	Ramp Enable

For more information on these command control codes, consult the device documentation.

**Examples**

1. To send the **Fast Stop** command to the controller, use the syntax CC0.1 and write to the address.
2. To initiate a **Ramp Disable**, use the syntax CC0.14 and write to the address.

**M-Rotary Addressing**

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Long, DWord, <b>Float</b>	Read Only
Control Parameters	CP00-CP99	Long, DWord, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Fast Stop
02	Ramp Stop
03	Run
06	Batch Initiate

For more information on these command control codes, consult the device documentation.

**Examples**

1. To send the **Fast Stop** command to the controller, use the syntax CC0.1 and write to the address.

2. To initiate a **Batch Initiate**, use the syntax CC0.6 and write to the address.

### M-Shuttle Addressing

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Long, DWord, <b>Float</b>	Read Only
Control Parameters	CP00-CP99	Long, DWord, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Fast Stop
03	Run
04	Home Return
05	Home Seek
06	Home Set
07	Batch Reset
08	Setpoint Select 1
09	Setpoint Select 2
10	Setpoint Select 3
11	Setpoint Select 4

For more information on these command control codes, consult the device documentation.

### Examples

1. To send the **Fast Stop** command to the controller, use the syntax CC0.1 and write to the address.

2. To initiate a **Home Return**, use the syntax CC0.4 and write to the address.

### M-Track Addressing

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Long, DWord, <b>Float</b>	Read Only
Control Parameters	CP00-CP99	Long, DWord, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Fast Stop
02	Ramp Stop
03	Run
04	Enable Master Mode
05	Enable Follower Mode
06	Batch Initiate

For more information on these command control codes, consult the device documentation.

**Examples**

1. To send the **Fast Stop** command to the controller, use the syntax CC0.1 and write to the address.
2. To initiate a **Batch Initiate**, use the syntax CC0.6 and write to the address.

**M-Traversal Addressing**

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Long, DWord, <b>Float</b>	Read Only
Control Parameters	CP00-CP99	Long, DWord, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Fast Stop
03	Run
04	Home Return
05	Home Seek
06	Home Set
07	Batch Reset
08	Setpoint Select 1
09	Setpoint Select 2
10	Setpoint Select 3
11	Setpoint Select 4

For more information on these command control codes, consult the device documentation.

**Examples**

1. To send the **Fast Stop** command to the controller, use the syntax CC0.1 and write to the address.
2. To initiate a **Home Return**, use the syntax CC0.4 and write to the address.

**M-Trim Addressing**

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Short, Word, <b>Float</b>	Read Only
Control Parameters	CP00-CP99	Short, Word, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Fast Stop
02	Ramp Stop
03	Run

04	Enable Primary Mode
05	Enable Secondary Mode
06	Forward
07	Reverse
08	Enable Closed Loop
09	Enable Open Loop
10	Enable Primary Setpoint 1
11	Enable Primary Setpoint 2
12	Enable Secondary Setpoint 3
13	Enable Secondary Setpoint 4
14	Ramp Disable
15	Ramp Enable

For more information on these command control codes, consult the device documentation.

### Examples

1. To send the **Fast Stop** command to the controller, use the syntax CC0.1 and write to the address.
2. To initiate a **Ramp Disable**, use the syntax CC0.14 and write to the address.

### ML-Drive Addressing

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Short, Word, <b>Float</b>	Read Only
Control Parameters	CP00-CP99	Short, Word, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Fast Stop
02	Ramp Stop
03	Run
04	Enable Master Mode
05	Enable Follower Mode
10	Enable Setpoint 1
11	Enable Setpoint 2
12	Enable Setpoint 3
13	Enable Setpoint 4

For more information on these command control codes, consult the device documentation.

### Examples

1. To send the **Fast Stop** command to the controller, use the syntax CC0.1 and write to the address.
2. To initiate a **Ramp Stop**, use the syntax CC0.2 and write to the address.

### ML-Trim Addressing

The following addresses are valid. The default data types for dynamically defined tags are shown in **bold**.

Memory	Syntax	Data Type	Access
Monitor Variables	MV00-MV99	Short, Word, <b>Float</b>	Read Only

Control Parameters	CP00-CP99	Short, Word, <b>Float*</b>	Read/Write
Control Commands	CC0.Code	<b>Boolean</b>	Write Only

\*If a Float address has a bit appended (0 to 5 allowed), this will represent the Floating precision on any writes to the device.

**Note:** For more information on Floating point precision, refer to [Device Settings](#).

\*\*Valid Code: To execute these commands, write to the address as shown below.

Code	Description
01	Fast Stop
02	Ramp Stop
03	Run
04	Enable Master Mode
05	Enable Follower Mode
10	Enable Setpoint 1
11	Enable Setpoint 2
12	Enable Setpoint 3
13	Enable Setpoint 4

For more information on these command control codes, consult the device documentation.

**Examples**

1. To send the **Fast Stop** command to the controller, use the syntax CC0.1 and write to the address.
2. To initiate a **Ramp Stop**, use the syntax CC0.2 and write to the address.

## 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](#)

### 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](#)

### Device Specific Messages

[Unable to write to '<address>' on device '<device name>'. Contrex Error \(<error code>\) <error description>](#)

[Deactivating Tag '<address>' on device '<device name>'. Contrex Error \(<error code>\) <error description>](#)

## 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](#)

## 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**

---

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 to 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. Also verify 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:**

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 device.
2. 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](#)

## 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 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.

## Device Specific Messages

---

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

### Device Specific Messages

[Unable to write to '<address>' on device '<device name>.' Contrex Error \(<error code>\) <error description>](#)

[Deactivating Tag '<address>' on device '<device name>.' Contrex Error \(<error code>\) <error description>](#)

## Unable to write to '<address>' on device '<device name>.' Contrex Error (<error code>) <error description>

---

### 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 address that is being written to is invalid or the data is out of range.

### Solution:

1. Verify the cabling between the PC and the device.
2. Verify 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. Verify that the address is valid in the controller. Use the Contrex Error Description to determine the cause of the error.

### See Also:

[Error Descriptions](#)

## Deactivating Tag '<address>' on device '<device name>.' Contrex Error (<error code>) <error description>

---

### Error Type:

Serious

### Possible Cause:

An attempt has been made to reference a nonexistent location in the specified device. The driver will deactivate the tag.

### Solution:

Verify the tags assigned to addresses in the specified range on the device and eliminate ones that reference invalid locations. Use the Contrex Error Description that was returned to the driver to determine the cause of the error.

**See Also:**

[Error Descriptions](#)

# Index

## A

Address '<address>' is out of range for the specified device or register.....	15
Address Descriptions.....	8
Address Validation.....	14

## B

Boolean.....	7
--------------	---

## C

Cable Diagram.....	5
Communications error on '<channel name>' [<error mask>].....	16
COMn does not exist.....	15
COMn is in use by another application.....	16

## D

Data Type '<type>' is not valid for device address '<address>'.....	15
Data Types Description.....	7
Deactivating Tag '<address>' on device '<device name>' Contrex Error (<error code>) <error description>.....	17
Device '<device name>' is not responding.....	17
Device address '<address>' contains a syntax error.....	14
Device address '<address>' is not supported by model '<model name>'.....	15
Device address '<address>' is Read Only.....	15
Device ID.....	5
Device Setup.....	5
Device Specific Messages.....	17
Device Status Messages.....	16
DWord.....	7

**E**

Error Descriptions.....	14
Error opening COMn.....	16

**F**

Float.....	7
Framing.....	16

**L**

Long.....	7
-----------	---

**M**

M-Cut Addressing.....	8
M-Drive Addressing.....	8
M-Rotary Addressing.....	9
M-Shuttle Addressing.....	10
M-Track Addressing.....	10
M-Traverse Addressing.....	11
M-Trim Addressing.....	11
Mask.....	16
Missing address.....	14
ML-Drive Addressing.....	12
ML-Trim Addressing.....	12
Modem Setup.....	5

**N**

Network.....	5
--------------	---

**O**

Overrun..... 16  
Overview..... 4

**P**

Parity..... 16

**S**

Serial Communications..... 15  
Settings..... 6

**U**

Unable to set comm parameters on COMn..... 16  
Unable to write to '<address>' on device '<device name>.' Contrex Error (<error code>) ..... 17  
<error description>.....