

# **Toyopuc PC2 Serial Driver Help**

**© 2012 Kepware Technologies**

# Table of Contents

<b>Table of Contents</b> .....	<b>2</b>
<b>Toyopuc PC2 Serial Driver Help</b> .....	<b>3</b>
<b>Overview</b> .....	<b>3</b>
<b>Device Setup</b> .....	<b>4</b>
<b>Modem Setup</b> .....	<b>5</b>
<b>Diagnostics Tags</b> .....	<b>5</b>
<b>Multi Point Read Support</b> .....	<b>6</b>
<b>Data Types Description</b> .....	<b>8</b>
<b>Address Descriptions</b> .....	<b>9</b>
<b>Error Descriptions</b> .....	<b>10</b>
<b>Address Validation</b> .....	<b>10</b>
Missing address.....	10
Device address '<address>' contains a syntax error.....	10
Address '<address>' is out of range for the specified device or register.....	11
Device address '<address>' is not supported by model '<model name>'.....	11
Data Type '<type>' is not valid for device address '<address>'.....	11
Device address '<address>' is Read Only.....	11
Array size is out of range for address '<address>'.....	11
Array support is not available for the specified address: '<address>'.....	11
<b>Serial Communications</b> .....	<b>12</b>
COMn does not exist.....	12
Error opening COMn.....	12
COMn is in use by another application.....	12
Unable to set comm parameters on COMn.....	12
Communications error on '<channel name>' [<error mask>].....	13
<b>Device Status Messages</b> .....	<b>13</b>
Device '<device name>' is not responding.....	13
Unable to write to '<address>' on device '<device name>'.....	13
<b>Toyopuc PC2 Serial Device Specific Messages</b> .....	<b>13</b>
Bad address in block [<start address> to <end address>] on device '<device name>'.....	14
<b>Index</b> .....	<b>15</b>

## Toyopuc PC2 Serial Driver Help

---

Help version 1.013

### CONTENTS

#### [Overview](#)

What is the Toyopuc PC2 Serial Driver?

#### [Device Setup](#)

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

#### [Data Types Descriptions](#)

What data types does this driver support?

#### [Address Descriptions](#)

How do I address a data location on a Toyopuc PC2 Serial device?

#### [Error Descriptions](#)

What error messages does the Toyopuc PC2 Serial driver produce?

### Overview

---

The Toyopuc PC2 Serial Driver provides an easy and reliable way to connect Toyopuc PC2 Serial devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with any Toyoda Toyopuc PLC that supports Computer Link communications. The Toyopuc PC2 Serial driver supports extensive diagnostics tags and the Toyopuc PC2's Multi Point Read features. For more information, refer to [Diagnostics Tags](#) and [Multi Point Read Support](#).

**Device Setup**

**Supported Devices**

PC2 Series or any Computer Link compatible device

**Communication Protocol**

Computer Link

**Supported Communication Parameters**

Baud Rate: 300, 600, 1200, 2400, 9600, 19200

Parity: None, Even, or Odd

Data Bits: 5, 6, 7 or 8

Stop Bits: 1 or 2

**Note:** Not all devices support the listed configurations.

**Device IDs (Computer Link Station Number)**

The Device ID range is 0 to 37 Octal.

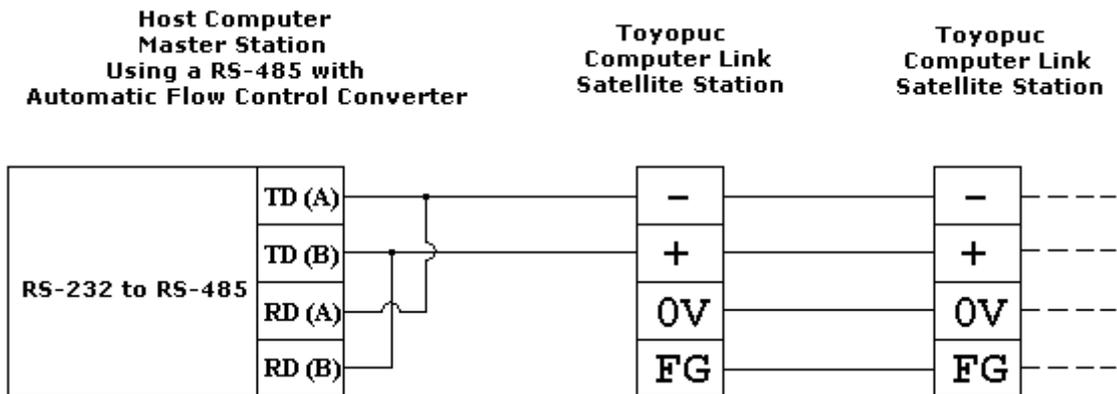
**Flow Control**

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 whereas others require RTS flow. To determine the converter's flow requirements, refer to its documentation. An RS485 converter that provides automatic flow control is recommended.

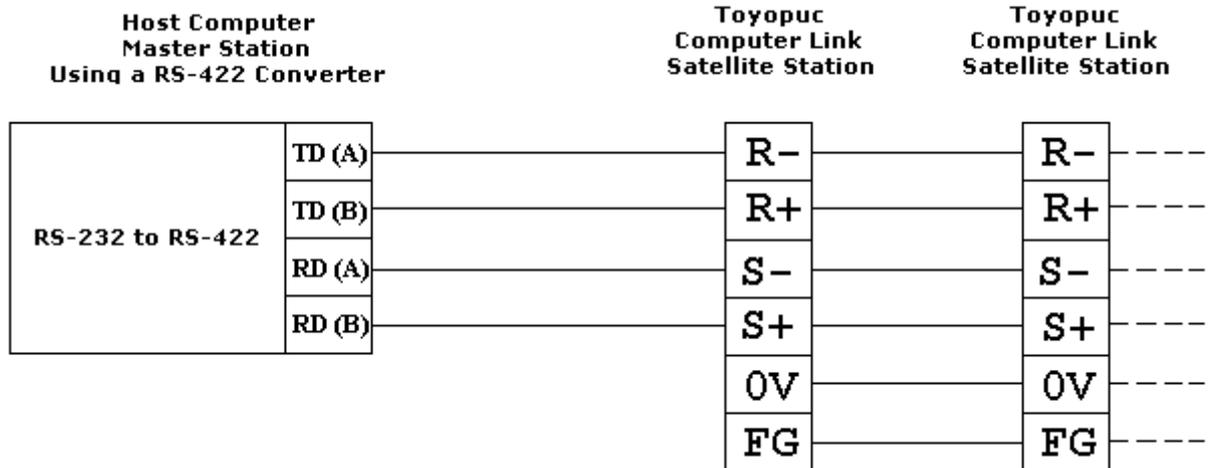
**Ethernet Encapsulation**

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

**Cable Connections**



**2 Wire Computer Link System**



#### 4 Wire Computer Link System

##### Modem Setup

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

##### Diagnostics Tags

Diagnostics Tags provide information on how the Toyopuc PC2 Serial driver is performing at both the channel and device level. At the channel level, the diagnostics tags provide information that covers all operations performed by the driver when communicating with any PLC on the network. At the device level, the diagnostics tags provide information that pertains only to the device under which the diagnostic tags have been requested.

##### Channel Level Diagnostics Tags

Tag Name	Functional Description
_ChannelReadTime	Contains the amount of time in milliseconds required to read all currently active data for all devices on this channel. This value is a signed long.
_ChannelHighTime	Contains the amount of time in milliseconds of longest read cycle. This value is a signed long.
_ChannelLowTime	Contains the amount of time in milliseconds of shortest read cycle. This value is a signed long.
_ChannelReadsPerformed	Contains a count of the reads performed on this channel for all devices. This is a signed long and will roll over.
_ChannelWritesPerformed	Contains a count of the writes performed on this channel for all devices. This is a signed long and will roll over.
_ChannelTimeouts	Contains a count of the number of timeout/message failures that have occurred for all devices. The _ChannelTimeouts count is for any error that may occur on a message attempt. The value does not necessarily indicate how many messages failed to be sent altogether. It should be used to diagnose possible communication issues with specific devices. This is a signed long and will roll over.

##### Device Level Diagnostics Tags

Tag Name	Functional Description
_DeviceReadTime	Contains the amount of time in milliseconds required to read a block of data from the specified device. This value is a signed long.
_DeviceHighTime	Contains the longest amount of time in milliseconds required to read a block of data from the specified device. This value is a signed long.
_DeviceLowTime	Contains the shortest amount of time in milliseconds required to read a block of data from the specified device. This value is a signed long.
_DeviceReadsPerformed	Contains a count of the reads performed on this device. This is a signed long and will roll over.
_DeviceWritesPerformed	Contains a count of the writes performed on this device. This is a signed long and will roll over.
_DeviceTimeouts	Contains a count of the number of timeout/message failures that have occurred on the specified device. The _DeviceTimeouts count is for any error that may occur on a message attempt. The value does not necessarily indicate how many messages failed to be sent altogether. It should be used to diagnose possible communication issues with this specific device. This is a signed long.

	and will roll over.
_DeviceMultiPointReads	Contains a count of the number of Multi Point read requests that are currently being used to acquire all data that is marked for Multi Point operation. This tag can be used to tune Multi Point read operation. The goal of course being to limit the number of Multi Point reads being done to the lowest count possible, preferably 1. This is a signed long and will roll over.

**Note:** All diagnostics tags are Read/Write. The only value that can be written to the tags is zero (which will clear or reset them).

### Multi Point Read Support

The Toyopuc PLC supports the ability to read data spread randomly throughout the PLC using a single command. This Multi Point command is limited to reading 128 individual 16 bit values in a single request. By using this command, users can read crucial data items quickly and efficiently. The Toyopuc PC2 Serial driver automatically attempts to make the use of the Multi Point command both easy and efficient. Any memory type that can be acquired by the Toyopuc PC2 Serial driver can be part of a Multi Point read command. To mark a particular data item to be part of a Multi Point request simply place the '#' character in front of any current memory type. The table below is shown with the addition of the '#' character to each memory type.

There are some things that should be considered when using the Multi Point read functions. As stated, the Multi Point function can only read 128 16 bit values per request. With this in mind the Multi Point command can increase the speed of data acquisition but if over used the driver will still need to make multiple Multi Point commands to read all the requested data. When this occurs, the overall performance of the driver would again be reduced. The key is to use the Multi Point command wisely.

The driver will automatically group data from memory types like bit memory into 16 bit values. For example if X1, X3, X4, X6, X9, XA, XB are marked as part of a Multi Point read using the '#,' enter an address of #X1, #X3, #X4, #X6, #X9, #XA, #XB, so that these 7 items would be placed into a single 16 bit value. Thus, they would only use one of the 128 16 bit values available in a single Multi Point read command. The 7 items were grouped together because the address of each bit fell within a single 16 bit word value of X memory. If 7 items like #X1, #X20, #X55, #X77, #X99, #XAA, #XBB are entered as part of a Multi Point read, each bit in this case would require an entire 16 bit value in the Multi Point read command to receive the data. Plan the data usage in the controller. If possible make sure that the bits being read are grouped closely. This prudent planning applies primarily to the bit memory types. Register memory requires a single 16 bit value for each register added to the Multi Point read.

Such information will help plan Multi Point reads. The Toyopuc PC2 Serial driver can perform as many Multi Point reads as are needed to acquire all the data that has been have marked for Multi Point operation, but the more reads done, the slower the driver will be. To determine how many Multi Point read request the Toyopuc PC2 Serial driver is using to acquire all currently defined Multi Point data, use the special diagnostic tag "\_Device-MultiPointReads." For more information, refer to [Diagnostics](#).

The Multi Point read operation can be combined with the normal data reads of the Toyopuc PC2 Serial driver. If, for example, a block of 50 D registers consecutively ordered is being read, it may be more efficient to read the 50 D registers as part of a normal block read and save the space in the Multi Point read function for data that is spread more randomly throughout the PLC memory. Use the diagnostic tags to help determine the application's best method of acquiring data.

### Multi Point Limitations

For tags belonging to the PC2 series device model, the maximum data requested for the data types are as follows:

- Boolean: 2048\*
- Byte: 256
- Word: 128
- DWord: 64

\*If contiguous Booleans are requested, the request will be done in one Multi Point read.

Combination of any of the above data types in a single multi request has to be within the following limit:

$$(No. of Booleans/16) + (No. of Bytes/2) + No. of Words + (No. of DWords * 2) <= 128$$

### Memory Types Shown with the Multi Point Marker

Memory Type	Syntax	Data Types	Access
Edge Relay (P)	#P0000-#P01FF	Boolean	Read/Write

	#P000-#P01F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Keeping Relay (K)	#K0000-#K02FF	<b>Boolean</b>	Read/Write
	#K000-#K02F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Specific Relay (V)	#V0000-#V00FF	<b>Boolean</b>	Read/Write
	#V000-#V0F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Timer Bits (T)	#T0000-#T01FF	<b>Boolean</b>	Read/Write
	#T000-#T01F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Count Bits (C)	#C0000-#C01FF	<b>Boolean</b>	Read/Write
	#C000-#C01F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Link Relay (L)	#L0000-#L07FF	<b>Boolean</b>	Read/Write
	#L000-#L07F	Byte, Word, Short, BCD, DWord, Long, LBCD	
I/O Relay (X)	#X0000-#X07FF	<b>Boolean</b>	Read/Write
	#X000-#X07F	Byte, Word, Short, BCD, DWord, Long, LBCD	
I/O Relay (Y)	#Y0000-#Y07FF	<b>Boolean</b>	Read/Write
	#Y000-#Y07F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Internal Relay (M)	#M0000-#M07FF	<b>Boolean</b>	Read/Write
	#M000-#M07F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Specific Register (S)	#S0000-#S03FF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write
Timer/Counter Value (N)	#N0000-#N01FF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write
Link Register (R)	#R0000-#R07FF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write
Data Register (D)	#D0000-#D2FFF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write
File Register (B)	#B0000-#B1FFF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write

## Data Types Description

---

Data Type	Description
Boolean	Single bit
Byte	Unsigned 8 bit value bit 0 is the low bit bit 7 is the high 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.

## Address Descriptions

The Toyopuc PC2 Series supports the following addresses. The default data types for dynamically defined tags are shown in **bold**.

Memory Type	Syntax	Data Types	Access
Edge Relay (P)	P0000-P01FF	<b>Boolean</b>	Read/Write
	P000-P01F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Keeping Relay (K)	K0000-K02FF	<b>Boolean</b>	Read/Write
	K000-K02F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Specific Relay (V)	V0000-V00FF	<b>Boolean</b>	Read/Write
	V000-V0F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Timer Bits (T)	T0000-T01FF	<b>Boolean</b>	Read/Write
	T000-T01F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Count Bits (C)	C0000-C01FF	<b>Boolean</b>	Read/Write
	C000-C01F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Link Relay (L)	L0000-L07FF	<b>Boolean</b>	Read/Write
	L000-L07F	Byte, Word, Short, BCD, DWord, Long, LBCD	
I/O Relay (X)	X0000-X07FF	<b>Boolean</b>	Read/Write
	X000-X07F	Byte, Word, Short, BCD, DWord, Long, LBCD	
I/O Relay (Y)	Y0000-Y07FF	<b>Boolean</b>	Read/Write
	Y000-Y07F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Internal Relay (M)	M0000-M07FF	<b>Boolean</b>	Read/Write
	M000-M07F	Byte, Word, Short, BCD, DWord, Long, LBCD	
Specific Register (S)	S0000-S03FF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write
Timer/Counter Value (N)	N0000-N01FF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write
Link Register (R)	R0000-R07FF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write
Data Register (D)	D0000-D2FFF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write
File Register (B)	B0000-B1FFF	Byte, <b>Word</b> , Short, BCD, DWord, Long, LBCD	Read/Write

### Multi Point Read Support

The Multi Point read support allows the Toyopuc PC2 Serial driver to read data from multiple memory types in a single request. For more information, refer to [Multi Point Read Support](#).

### Diagnostics Tags

The Diagnostics Tags provide information on how the Toyopuc PC2 Serial driver is performing. For more information, refer to [Diagnostics Tags](#).

### Low/High Byte Modifier

An optional Low (L) or High (H) byte modifier can be appended to any address. This modifier instructs the driver that the low or high byte of the address word is requested.

**Note:** When adding a static tag with a low/high byte modifier the data type must be set to Byte. For more information on static vs. dynamic tags, refer to the OPC Server's online help.

### Examples

1. Request 'Specific Relay 100'--> V100.
2. Request High Byte of 'Data Register 10'--> D10H.
3. Request Long Value (2 consecutive 16 bit registers) starting at 'Link Register 7F'--> L7F@LONG (set datatype to Long for static tags, or append '@LONG' to address for dynamic tags).

## 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>'](#)

### Toyopuc PC2 Serial Device Specific Messages

[Bad address in block \[<start address> to <end address>\] 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 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 that 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 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 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.

---

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

---

**Error Type:**

Warning

**Possible Cause:**

A tag address that has been specified dynamically 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 dynamically 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 in the Channel Properties.

**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.

**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 communication parameters for the serial connection are incorrect.

### **Solution:**

1. Verify the cabling between the PC and the device.
2. Verify that the specified communication 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>'](#)

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

## **Toyopuc PC2 Serial Device Specific Messages**

---

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

**Toyopuc PC2 Serial Device Specific 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. Eliminate ones that reference invalid locations.

# Index

## A

Address '<address>' is out of range for the specified device or register.....	11
Address Descriptions.....	9
Address Validation.....	10
Array size is out of range for address '<address>'.....	11
Array support is not available for the specified address: '<address>'.....	11

## B

Bad address in block [<start address> to <end address>] on device '<device name>'.....	14
BCD.....	8
Boolean.....	8

## C

Communications error on '<channel name>' [<error mask>].....	13
COMn does not exist.....	12
COMn is in use by another application.....	12

## D

Data Type '<type>' is not valid for device address '<address>'.....	11
Data Types Description.....	8
Device '<device name>' is not responding.....	13
Device address '<address>' contains a syntax error.....	10
Device address '<address>' is not supported by model '<model name>'.....	11
Device address '<address>' is Read Only.....	11
Device ID.....	4
Device Setup.....	4
Device Status Messages.....	13
Diagnostics Tags.....	5
DWord.....	8

**E**

**Error Descriptions..... 10**  
**Error opening COMn..... 12**

**F**

**Framing..... 13**

**L**

**LBCD..... 8**  
**Long..... 8**

**M**

**Mask..... 13**  
**Missing address..... 10**  
**Modem Setup..... 5**  
**Multi Point Read Support..... 6**

**N**

**Network..... 4**

**O**

**Overrun..... 13**  
**Overview..... 3**

**P**

Parity..... 13

**S**

Serial Communications..... 12

Short..... 8

**T**

Toyopuc PC2 Serial Device Specific Messages..... 13

**U**

Unable to set comm parameters on COMn..... 12

Unable to write tag '<address>' on device '<device name>'..... 13

**W**

Word..... 8