# Table of Contents

1 Getting Started................................................................................................................................................. 2
   Help Contents.................................................................................................................................................. 2
   Overview ....................................................................................................................................................... 2

2 Device Setup.................................................................................................................................................... 2
   Device Setup .................................................................................................................................................. 2
   Modem Setup .................................................................................................................................................. 3
   Protocol Options ........................................................................................................................................... 3
   String Options............................................................................................................................................... 4

3 Data Types Description...................................................................................................................................... 4
   Data Types Description.................................................................................................................................. 4

4 Address Descriptions........................................................................................................................................ 4
   Address Descriptions..................................................................................................................................... 4

5 Error Descriptions............................................................................................................................................ 8
   Error Descriptions......................................................................................................................................... 8
   Address Validation.......................................................................................................................................... 8
   Address Validation.......................................................................................................................................... 8
      Address Validation...................................................................................................................................... 8
      Missing address .......................................................................................................................................... 9
      Device address '<address>' contains a syntax error ..................................................................................... 9
      Address '<address>' is out of range for the specified device or register ....................................................... 9
      Device address '<address>' is not supported by model '<model name>' ...................................................... 9
      Data Type '<type>' is not valid for device address '<address>' ................................................................... 9
      Array size is out of range for address '<address>' ..................................................................................... 10
      Array support is not available for the specified address: '<address>'......................................................... 10

Serial Communications........................................................................................................................................ 10
   Serial Communications................................................................................................................................. 10
   COMn does not exist....................................................................................................................................... 10
   Error opening COMn...................................................................................................................................... 10
   COMn is in use by another application.......................................................................................................... 11
   Unable to set comm parameters on COMn.................................................................................................... 11
   Communications error on COMn [<error mask>]............................................................................................... 11

Device Status Messages.................................................................................................................................... 11
   Device Status Messages................................................................................................................................. 11
   Device '<device name>' is not responding...................................................................................................... 11
   Unable to write to '<address>' on device '<device name>' ........................................................................... 12

Driver Warning Messages.................................................................................................................................. 12
   Driver Warning Messages............................................................................................................................. 12
   <Read or Write> request returned error code '<error code hex (decimal)>' for address '<address>' .......... 12
   Bad block starting at '<address>' for a length of '<bytes>' on device '<device name>' ................................ 12
   Initialization conflict error ............................................................................................................................ 13

Index .................................................................................................................................................................. 14
Siemens S5 (3964R) Driver Help

Help version 1.004

CONTENTS

Overview
What is the Siemens S5 (3964R) 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 Siemens S5 (3964R) device?

Error Descriptions
What are the error messages for the Siemens S5 (3964R) driver?

Overview
The Siemens S5 (3964R) Driver was designed specifically for use with 32 bit OPC Server products running on Intel microprocessor based computers. It is intended for use with Siemens S5 PLCs communicating via a communications processor card (such as the CP 544) configured to use the 3964R, or 3964 protocols and the RK 512 computer link. Multiple CPU systems are supported. This driver is not designed to respond to unsolicited data from the PLC. Refer to the Server Help documentation for operating system (OS) requirements.

Device Setup
Supported Devices
Any device that supports 3964 or 3964R protocol and uses the RK 512 computer link program.

Communication Protocol
3964R and 3964

Note: The 3964 variant is identical to 3964, except it does not use a Byte Check Character.

Supported Communication Parameters
Values depend on the communications processor card used and its configuration. The following values are typical settings.
Baud: 300 to 19200
Parity: Even
Data Bits: 8
Stop Bits: 1

Device Configuration
The device must be configured to operate in slave mode, with low priority partner.

Unsolicited Messages
This driver will accept and acknowledge unsolicited messages from the PLC, but does not use them. For optimum driver performance, unsolicited messages are discouraged.

Cable Connections
Communication processor cards that are compatible with the S5 PLC offer many connection options. A typical setup using a CP 525 card and the V.24 sub module is illustrated below. For more information, refer to the communication processor card's documentation.
Device Properties
There are several driver configuration options that depend on the communication processor card configuration. These include the protocol variation (3964 or 3964R), multiple CPU addressing, and communication processor card/CPU handshaking options. These may be set in the Protocol Options dialog of the server’s Device Wizard. The String Options dialog can also be used to specify how string data is passed to client applications.

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

Protocol Options
The Protocol Options dialog is used to specify the protocol variation, CPU number, handshaking flag parameters and byte order.

NOTE:
Do not connect the signal ground and shield together.
This could damage the submodule.
Most device configurations utilize the 3964R protocol version, which uses a Byte Check Character as a means of communication error checking. In rare cases, the 3964 protocol version (which is identical to 3964R but doesn't use the Byte Check Character) may be utilized.

If the system has multiple CPUs, the message packets must specify the CPU with which to communicate. To do so, check the **Specify CPU number in packet** box and enter the CPU number in the box below it. If only one CPU is used, make sure to leave the **Specify CPU number in packet** box unchecked.

If the configuration requires the message packets to specify handshaking flag parameters, check the **Specify handshaking flag in packet** box. Then, enter the flag byte and flag bit in the boxes below it.

The final option is used to specify byte ordering. Normally, Siemens devices will store multi-byte data values from least to most significant byte (LO to HI). If the device is programmed to store data in the opposite order, check **Reverse byte order** at the bottom of the dialog. This setting does not affect string data.

### String Options

The string options page is used to specify how string data is passed to client applications. The driver always reads all of the bytes in the memory range specified by a string tag's address.

If the **Pass full length strings to clients** box is left unchecked (the default setting) strings sent to the clients will be terminated at the first null character (0x00) encountered in the data. If there are no null characters in the data, then one will be placed at the end of the passed string.

If the box is checked, then all of the characters contained in the memory range will be sent to the client applications. All null characters in the data will be replaced with a user specified character to allow the client application to display the full range of data. The replacement character is specified with the **Replace nulls with** edit box. The number entered here must be in hex. For example, 20 would result in nulls being replaced by spaces, 2A would use the asterisk character, and etc. A null character will be placed at the end of the string.

### Data Types Description

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>Single bit of an 8 bit value</td>
</tr>
<tr>
<td>Byte</td>
<td>Unsigned 8 bit value</td>
</tr>
<tr>
<td>Word</td>
<td>Unsigned 16 bit value</td>
</tr>
<tr>
<td>Short</td>
<td>Signed 16 bit value</td>
</tr>
<tr>
<td>DWord</td>
<td>Unsigned 32 bit value</td>
</tr>
<tr>
<td>Long</td>
<td>Signed 32 bit value</td>
</tr>
<tr>
<td>Float</td>
<td>32 bit floating point value</td>
</tr>
<tr>
<td>BCD</td>
<td>Two byte packed BCD</td>
</tr>
<tr>
<td></td>
<td>Value range is 0-9999. Behavior is undefined for values beyond this range.</td>
</tr>
<tr>
<td>LBCD</td>
<td>Four byte packed BCD</td>
</tr>
<tr>
<td></td>
<td>Value range is 0-99999999. Behavior is undefined for values beyond this range.</td>
</tr>
<tr>
<td>String</td>
<td>Null terminated ASCII string</td>
</tr>
</tbody>
</table>

*For more information, refer to [Address Descriptions](#).*

### Address Descriptions

Address specifications vary depending on the device in use. If you attempt to access a data element that goes beyond
the range of your device the server will generate and error message to alert you to this condition. The default data types for dynamically defined tags are shown in **bold**.

<table>
<thead>
<tr>
<th>Device Type</th>
<th>Range Data</th>
<th>Type</th>
<th>Array Access</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete Inputs</td>
<td>I0.b-I255.b</td>
<td><strong>Boolean</strong></td>
<td>No</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>.b is Bit Number 0-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>IB0-IB255</td>
<td><strong>Byte</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>IWO-IW254</td>
<td><strong>Word, Short</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>ID0-ID252</td>
<td><strong>DWord, Long</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td>Discrete Inputs</td>
<td>E0.b-E255.b</td>
<td><strong>Boolean</strong></td>
<td>No</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>.b is Bit Number 0-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EB0-EB255</td>
<td><strong>Byte</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>EW0-EW254</td>
<td><strong>Word, Short</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>ED0-ED252</td>
<td><strong>DWord, Long</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td><strong>Note:</strong> I and E access the same memory area.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Discrete Outputs</td>
<td>Q0.b-Q255.b</td>
<td><strong>Boolean</strong></td>
<td>No</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>.b is Bit Number 0-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>QB0-QB255</td>
<td><strong>Byte</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>QW0-QW254</td>
<td><strong>Word, Short</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>QD0-QD252</td>
<td><strong>DWord, Long</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td>Discrete Outputs</td>
<td>A0.b-A255.b</td>
<td><strong>Boolean</strong></td>
<td>No</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>.b is Bit Number 0-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AB0-AB255</td>
<td><strong>Byte</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>AW0-AW254</td>
<td><strong>Word, Short</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>AD0-AD252</td>
<td><strong>DWord, Long</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td><strong>Note:</strong> Q and A access the same memory area.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internal Memory</td>
<td>F0.b-F255.b</td>
<td><strong>Boolean</strong></td>
<td>No</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>.b is Bit Number 0-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FB0-FB255</td>
<td><strong>Byte</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>FW0-FW254</td>
<td><strong>Word, Short, BCD</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>FD0-FD252</td>
<td><strong>DWord, Long, LBCD</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td>Internal Memory</td>
<td>M0.b-M255.b</td>
<td><strong>Boolean</strong></td>
<td>No</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>.b is Bit Number 0-7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>MB0-MB255</td>
<td><strong>Byte</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>MW0-MW254</td>
<td><strong>Word, Short, BCD</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td></td>
<td>MD0-MD252</td>
<td><strong>DWord, Long, LBCD</strong></td>
<td>Yes</td>
<td>Read Only</td>
</tr>
<tr>
<td><strong>Note:</strong> F and M access the same memory area.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Block</td>
<td><strong>DBn:KM0.b-KM255.b</strong></td>
<td><strong>Boolean</strong></td>
<td>No</td>
<td>Read/Write</td>
</tr>
<tr>
<td>Boolean</td>
<td>n is Block Number (1-255)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.b is Bit Number 0-15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Block</td>
<td><strong>DBn:KL0-KL255</strong></td>
<td><strong>Byte</strong></td>
<td>No</td>
<td>Read/Write</td>
</tr>
<tr>
<td>Left Byte</td>
<td>n is Block Number (1-255)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Block</td>
<td><strong>DBn:KR0-KR255</strong></td>
<td><strong>Byte</strong></td>
<td>No</td>
<td>Read/Write</td>
</tr>
</tbody>
</table>

www.kepware.com
<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Byte</td>
<td><strong>n</strong> is Block Number (1-255)</td>
</tr>
<tr>
<td>Data Block</td>
<td><strong>DBn</strong>:KH0-KH255</td>
</tr>
<tr>
<td><strong>Word</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Read/Write</strong></td>
<td></td>
</tr>
<tr>
<td>Data Block</td>
<td><strong>DBn</strong>:KF0-KF255</td>
</tr>
<tr>
<td><strong>Short</strong>, <strong>BCD</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Read/Write</strong></td>
<td></td>
</tr>
<tr>
<td>Signed Word</td>
<td><strong>DBn</strong>:KD0-KD254</td>
</tr>
<tr>
<td><strong>Long</strong>, <strong>DWord</strong>, <strong>LBCD</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Read/Write</strong></td>
<td></td>
</tr>
<tr>
<td>Signed Long</td>
<td><strong>DBn</strong>:KG0-KG254</td>
</tr>
<tr>
<td>Float</td>
<td><strong>DBn</strong>:KS0.0-NS255.1</td>
</tr>
<tr>
<td>String</td>
<td><strong>DBn</strong>:KT0-KT255</td>
</tr>
<tr>
<td>S5 Timer as DB</td>
<td><strong>DBn</strong>:KC0-KC255</td>
</tr>
<tr>
<td><strong>Word</strong>, <strong>Short</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Read/Write</strong></td>
<td></td>
</tr>
<tr>
<td>S5 Counter as DB</td>
<td><strong>DBn</strong>:KM0.0-KM255.b</td>
</tr>
<tr>
<td><strong>Boolean</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Read/Write</strong></td>
<td></td>
</tr>
<tr>
<td>Extended Data Block</td>
<td><strong>DBn</strong>:KL0-KL255</td>
</tr>
<tr>
<td>Left Byte</td>
<td><strong>DBn</strong>:KR0-KR255</td>
</tr>
<tr>
<td><strong>Byte</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Read/Write</strong></td>
<td></td>
</tr>
<tr>
<td>Extended Data Block</td>
<td><strong>DBn</strong>:KH0-KH255</td>
</tr>
<tr>
<td>Right Byte</td>
<td><strong>DBn</strong>:KF0-KF255</td>
</tr>
<tr>
<td><strong>Word</strong>, <strong>Short</strong>, <strong>BCD</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Read/Write</strong></td>
<td></td>
</tr>
<tr>
<td>Extended Data Block</td>
<td><strong>DBn</strong>:KD0-KD254</td>
</tr>
<tr>
<td>Signed Word</td>
<td><strong>DBn</strong>:KG0-KG254</td>
</tr>
<tr>
<td>Extended Data Block</td>
<td><strong>DBn</strong>:KS0.0-NS255.1</td>
</tr>
<tr>
<td>String</td>
<td><strong>DBn</strong>:KT0-KT255</td>
</tr>
<tr>
<td>S5 Timer as DB</td>
<td><strong>DBn</strong>:KC0-KC255</td>
</tr>
<tr>
<td><strong>Word</strong>, <strong>Short</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Read/Write</strong></td>
<td></td>
</tr>
</tbody>
</table>
Timer Current Values | T0-T255 | Long | Yes | Read Only
--- | --- | --- | --- | ---
Counter Current Values | C0-C255 | Word, Short | Yes | Read Only
Counter Current Values | Z0-Z255 | Word, Short | Yes | Read Only

**Note:** All offsets for memory types I, Q, and F represent a byte starting location within the specified memory type.

**Examples**

1. To access bit 3 of Internal Memory F20, declare an address as follows:
   F20.3

2. To access Data Block 5 as word memory at element 30 declare an address as follows:
   DB5:KH30

3. To access Data Block 2 element 20 and bit 7 declare an address as follows:
   DB2:KM20.7

4. To access Data Block 1 as left byte memory at element 10 declare an address as follows:
   DB1:KL10

5. To access Internal Memory F20 as a DWORD, declare an address as follows:
   FD20

6. To access Input Memory I10 as a Word, declare an address as follows:
   IW10

**Note:** Use caution when modifying Word, Short, DWord, and Long types. For I, Q, and F each address starts at a byte offset within the device. Therefore, Words FW0 and FW1 overlap at byte 1. Writing to FW0 will also modify the value held in FW1. Similarly, DWord, and Long types can also overlap. It is recommended that you use these memory types so that overlapping does not occur. As an example, when using DWords you might want to use FD0, FD4, FD8 ... and so on, to prevent overlapping bytes.

**Timers**
The Siemens S5 3964R driver automatically scales T and KT values based on the Siemens S5 time format. The value returned for either a T or KT memory type will already be scaled using the appropriate Siemens time base. As a result the values are always returned as a count of milliseconds. When writing to T or KT memory types the Siemens time base will also be applied. To write a value to a timer in the controller, simply write the desired value as a count of milliseconds to the appropriate timer.

**Strings**
String data is stored in data block registers, thus the actual number of bytes used to store the data is an even number. For example, if a string of length 5 is specified, say by DB11:KS1.5, then 3 registers (6 bytes) will be used to store the string data. When writing strings shorter than the maximum specified length, 5 in our example, a null terminator (0x00) will be added to the end of the string. When strings are read, the full range of registers are read, 3 in our example. Use of string tags with overlapping address ranges should be avoided due to the effects of the null terminators. See **String Options** for more information on how strings can be formatted.

**Arrays**
Arrays are supported for the memory types indicated in the table above. An array can be declared using the following syntax:

```
<address> [rows] [cols]
<address> [cols] with an assumed row count of 1.
```

The maximum size of an array is 128 bytes, where the size of an array is calculated as:

```
size = rows * cols * (data type size in bytes).
```

The data type size in bytes is: 1 for Byte; 2 for Word, and Short; and 4 for DWord, Long, and Float. Timers are an exception, where a data size of 2 bytes should be used.

All locations referenced by an array must exist in the device. If this is not the case, the device will indicate an invalid address upon read or write and the driver will deactivate the tag. For example, if data block 20 has a size of 10 words (KH0 to HK9), then:

```
DB20:KH1 [4] is valid. Element 1 references KH1, element 2 references KH2, element 3 references KH3, and element 4 references KH4.
```
DB20:KG1 [4] would be valid. Element 1 references KH1 and KH2, element 2 references KH3 and KH4, element 3 references KH5 and KH6, and element 4 references KH7 and KH8.

DB20:KH8 [4] is invalid. Element 1 references KH8, element 2 references KH9, element 3 references KH10, and element 4 references KH11. The last 2 elements reference nonexistent locations.

Counter address range from C0 to C255. Therefore:

C1 [4] is valid.

C253 [4] is invalid because the last element references the nonexistent counter C256.

**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>'
- 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 COMn [<error mask>]

**Device Status Messages**

- Device '<device name>' is not responding
- Unable to write to '<address>' on device '<device name>'

**Driver Warning Messages**

- <Read or Write> request returned error code '<error code hex (decimal)>' for address '<address>' on device '<device name>'
- Bad block stating at '<address>' for a length of '<bytes>' on device '<device name>'. The block has been deactivated
- Initialization conflict error

**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>'
- 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 via DDE 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.
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 COMn [<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 COMn [<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
- 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 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 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.

**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 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.

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

**Driver Warning Messages**

<Read or Write> request returned error code '<error code hex (decimal)>' for address '<address>' on device '<device name>'

Bad block starting at '<address>' for a length of '<bytes>' on device '<device name>'. The block has been deactivated

Initialization conflict error

<Read or Write> request returned error code '<error code hex (decimal)>' for address '<address>' on device '<device name>'

**Error Type:**
Serious

**Possible Cause:**
Consult Siemens RK512 computer link documentation for meaning of "RETEL" error code.

**Solution:**
Consult Siemens RK512 computer link documentation for meaning of "RETEL" error code.

**Note:**
By allowing the device to timeout, the driver resolves an "Out of Sync" condition (RETEL code 0x36). Thus, it is normal for this message to be followed by a "Device not responding" error.

Bad block starting at '<address>' for a length of '<bytes>' on device '<device name>'. The block has been deactivated

**Error Type:**
Serious

**Possible Cause:**
The device has been configured in such a way as to make one or more of the addresses within the block inaccessible.

**Solution:**
1. Use different addresses.
2. Alter the device configuration.

### Initialization conflict error

**Error Type:**
Warning

**Possible Cause:**
The device made an attempt to send the driver an unsolicited message at the same time the driver is making a read or write request. This indicates that your device may not be properly configured as the "low priority partner", or is programmed to send unsolicited data. This message may also result from the normal repetition of a packet by the device due to a transient communication problem. Occasional instances of this warning are not necessarily cause for concern.

**Solution:**
Make sure that the device is configured as the "low priority partner" and is not programmed to send unsolicited data.
Index

- < -
<Read or Write> request returned error code
‘<error code hex (decimal)>’ for address
‘<address>’ on device ‘<device name>’

- A -
Address ‘<address>’ is out of range for the
specified device or register
Address Descriptions
Address Validation
Array size is out of range for address ‘<address>
Array support is not available for the specified
address: ‘<address>

- B -
Bad block starting at ‘<address>’ for a length of
‘<bytes>’ on device ‘<device name>’. The block
has been deactivated
bits
Boolean
Byte

- C -
Cable Connections
Communications error on COMn [<error mask>]
COMn does not exist
COMn is in use by another application

- D -
data blocks
Data Type ‘<type>’ is not valid for device address
‘<address>’
Data Types Description
Device ‘<device name>’ is not responding
Device address ‘<address>’ contains a syntax error

- E -
Error Descriptions
Error opening COMn.

- F -
flags
Float

- I -
Initialization conflict error
internal

- L -
Long

- M -
Master ID
memory
Missing address
Modem Setup

- N -
Network

- O -
Overview

- P -
Protocol Options
- S -
Serial Communications  10
  Short  4
  String Options  4

- T -
tags  4

- U -
Unable to set comm parameters on COMn  11
Unable to write tag '<address>' on device '<device name>'  12

- W -
Word  4
words  4