

Siemens S7 MPI Driver Help

© 2012 Kepware Technologies

Table of Contents

Table of Contents	2
Siemens S7 MPI Driver Help	3
Overview	3
Device Setup	4
Modem Setup	5
Data Types Description	6
S7-300 & S7-400 Address Descriptions	7
Error Descriptions	9
Address Validation	9
Missing address.....	9
Device address '<address>' contains a syntax error.....	9
Address '<address>' is out of range for the specified device or register.....	10
Device address '<address>' is not supported by model '<model name>'.....	10
Data Type '<type>' is not valid for device address '<address>'.....	10
Device address '<address>' is Read Only.....	10
Serial Communications	10
COMn does not exist.....	10
Error opening COMn.....	11
COMn is in use by another application.....	11
Unable to set comm parameters on COMn.....	11
Communications error on '<channel name>' [<error mask>].....	11
Device Status Messages	11
Device '<device name>' is not responding.....	12
Unable to write to '<address>' on device '<device name>'.....	12
Connection timeout on MPI node <device id>.....	12
Request timeout on MPI node <device id>.....	12
Device Specific Messages	13
Bad address in block '<start address>' to '<end address>' on device '<device name>', the block has been deactivated.....	13
Index	14

Siemens S7 MPI Driver Help

Help version 1.015

CONTENTS

[Overview](#)

What is the Siemens S7 MPI 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 S7 300/400 device?

[Error Descriptions](#)

What error messages does the Siemens S7 MPI driver produce?

Overview

The Siemens S7 MPI Driver provides an easy and reliable way to connect Siemens S7 MPI devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with Siemens S7 300 and 400 PLCs communicating via the MPI interface.

The Siemens S7 MPI serial port adapter handles MPI connectivity on the PC. This adapter may be obtained from Siemens using the following part numbers:

Siemens Part: 6ES7-972-OCA23-OXAO Version 5.1

Siemens Part: 6ES7-972-OCA22-OXAO Version 5.0

This is just a partial list of the part numbers for more recent versions of this adapter. This driver has been developed to work with versions of the adapter that predate version 5.0 or 5.1.

The Siemens S7 MPI PC Adapter and the S7 MPI driver have been designed to allow more than one master device to exist on the network. This allows a programming package to operate while this driver is running or while additional PCs are accessing the network.

Device Setup

Supported Devices

Siemens S7-300 Devices
Siemens S7-400 Devices

Communication Protocol

Multi Point Interface (MPI) S7-300/400 Communications Protocol

Supported Communication Parameters

PC to S7 PC Adapter*

Baud Rate: Either 19200 or 38400 switch selectable on the PC adapter.**

Data Bits: 8

Parity: Odd

Stop Bits: 1

*S7 PC Adapter to S7-300/400 PLC.

**Baud Rate-187.5 K baud (fixed). The PC adapter is fixed at 187.5 K baud, which is the default baud rate of the MPI network.

Note: For proper operation, the baud rate selected in the S7 MPI driver must match the DIP switch selection on the S7 MPI PC adapter.

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 (such as the NetSLX). It may be invoked through the Communications dialog in Channel Properties. For more information, refer to the OPC server's help documentation.

Local Station Address

This parameter specifies the node number used by the Siemens S7 PC Adapter to determine its address on the MPI network. It must not conflict with any other node number used on the network. The valid local station addresses are 0 to 126. The default setting is 1.

Highest Station Address

This parameter specifies the highest MPI node that can exist on the network. It is required for proper network operation and is determined by the highest PLC node on the MPI network. There are four possible selections: 15, 31, 63 and 126. This selection can impact operation of the S7 MPI driver in two ways. If the value is too low, users may not be able to access PLCs with higher address settings. If the value is too high, the network performance may decrease as the S7 PC Adapter attempts to locate PLCs within a larger range than necessary. To provide the best network performance, S7 PLCs should start at address 3 and ascend consecutively. The default setting is 15.

PC is the Only Master

This parameter informs the driver that there may be more than one master present on the MPI network. This information is required for proper operation of the driver. For an application that requires direct peer-to-peer connection to a single PLC, this selection should be enabled. For a multi drop network in which multiple masters are present, this selection should be disabled. The default setting is enabled.

Device Station Addresses

Valid device addresses range from 0 to 126. Any devices defined under this channel should not use a device address that conflicts with this driver's Local Station address.

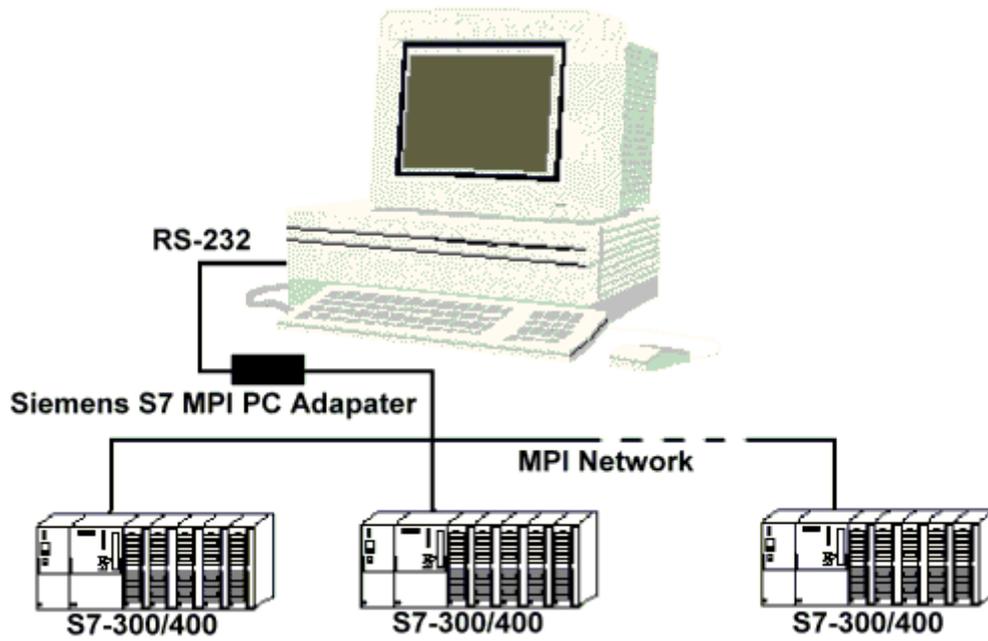
Cable Connections - PC to MPI Adapter

The PC to MPI adapter connection is accomplished using a null modem adapter with standard RS232 cable or a null modem cable. The cable provided by Siemens (P/N 6ES7 901-1BF00-OXA0) may also be used. It is a null modem cable, as well.

Cable Connections - MPI Adapter to PLC

The Siemens S7 PC Adapter allows quick and easy connection to a single S7-300/400 PLC. Although it is capable of multi drop operation, additional network wiring is required for proper operation. For more information on establishing a multi drop MPI network, refer to the Siemens S7 Hardware manual.

Note: Special consideration must be given to the PC Adapter wiring between the S7-300/400 PLCs and the Siemens S7 PC Adapter in a multi drop configuration (because the adapter is powered from the PLC's MPI port).



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	IEC 1131 Data Type
Boolean	Single bit of an 8 bit value*	BOOL
Byte	Unsigned 8 bit value	Byte
Char	Signed 8 bit value	Char
Word	Unsigned 16 bit value	Word
Short	Signed 16 bit value	INT
BCD	Two byte packed BCD Value range is 0-9999. Behavior is undefined for values beyond this range.	Word
DWord	Unsigned 32 bit value	DWORD
Long	Signed 32 bit value	DINT
LBCD	Four byte packed BCD Value range is 0-99999999. Behavior is undefined for values beyond this range.	DWORD
Float	32 bit floating point value The driver interprets two consecutive registers as a floating point value by making the second register the high word and the first register the low word.	REAL
String	Null terminated ASCII string**	STRING

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

**The Data Block subtype STRING is a null padded ASCII String.

S7-300 & S7-400 Address Descriptions

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

Address Type	Range	Data Type	Access
Discrete Inputs	I00000.b-I65535.b .b is Bit Number 0-7	Boolean	Read/Write
Discrete Inputs	IB00000-IB65535	Byte , Char, String*	Read/Write
Discrete Inputs	IW00000-IW65535	Word , Short, BCD	Read/Write
Discrete Inputs	ID00000-ID65535	DWord , Long, Float	Read/Write
Discrete Inputs	E00000.b-E65535.b .b is Bit Number 0-7	Boolean	Read/Write
Discrete Inputs	EB00000-EB65535	Byte , Char, String*	Read/Write
Discrete Inputs	EW00000-EW65535	Word , Short, BCD	Read/Write
Discrete Inputs	ED00000-ED65535	DWord , Long, Float	Read/Write
Note: I and E access the same memory area.			
Discrete Outputs	Q00000.b-Q65535.b .b is Bit Number 0-7	Boolean	Read/Write
Discrete Outputs	QB00000-QB65535	Byte , Char, String*	Read/Write
Discrete Outputs	QW00000-QW65535	Word , Short, BCD	Read/Write
Discrete Outputs	QD00000-QD65535	DWord , Long, Float	Read/Write
Discrete Outputs	A00000.b-A65535.b .b is Bit Number 0-7	Boolean	Read/Write
Discrete Outputs	AB00000-AB65535	Byte , Char, String*	Read/Write
Discrete Outputs	AW00000-AW65535	Word , Short, BCD	Read/Write
Discrete Outputs	AD00000-AD65535	DWord , Long, Float	Read/Write
Note: Q and A access the same memory area.			
Flag Memory	F00000.b-F65535.b .b is Bit Number 0-7	Boolean	Read/Write
Flag Memory	FB00000-FB65535	Byte , Char, String*	Read/Write
Flag Memory	FW00000-FW65535	Word , Short, BCD	Read/Write
Flag Memory	FD00000-FD65535	DWord , Long, Float	Read/Write
Flag Memory	M00000.b-M65535.b .b is Bit Number 0-7	Boolean	Read/Write
Flag Memory	MB00000-MB65535	Byte , Char, String*	Read/Write
Flag Memory	MW00000-MW65535	Word , Short, BCD	Read/Write
Flag Memory	MD00000-MD65535	DWord , Long, Float	Read/Write
Note: F and M access the same memory area.			
Data Block	DB1-N.DBX00000.b-DBX65535.b 1-N is DB Block Number .b is Bit Number 0-7	Boolean	Read/Write
Data Block	DB1-N.DBB00000-DBB65535 1-N is DB Block Number	Byte , Char, String*	Read/Write
Data Block	DB1-N.DBW00000-DBW65535 1-N is DB Block Number	Word , Short, BCD	Read/Write
Data Block	DB1-N.DBD00000-DBD65535 1-N is DB Block Number	DWord , Long, Float	Read/Write
Timer Current Values	T00000-T65535	DWord	Read Only
Counter Current Values	C00000-C65535	BCD , Word, Short	Read Only
Counter Current Values	Z00000-Z65535	BCD , Word, Short	Read Only

*Byte memory types (i.e. MB) support Strings. The syntax for strings is <address>. <length> where 0 < length <= 212.

Note: The actual number of addresses of each type depends on the Siemens S7-300 or S7-400 device in use. Each type does not necessarily support an address of 0 to 65535. For a list of all address ranges, refer to the device's documentation. All offsets represent a byte starting location within the specified memory type.

Arrays

All memory types and subtypes support arrays, except for Boolean data types, timers and counters. The valid syntax for declaring an array is as shown below. If no rows are specified, a row count of 1 is assumed.

`<address>[rows][cols]`

For Word, Short, BCD and "UBCD" arrays, the base address + (rows * cols * 2) cannot exceed 65536. Keep in mind that the elements of the array are words, located on a word boundary. For example, AW0[4] would return AW0, AW2, AW4 and AW6.

For Float, DWord, Long, Long BCD, "KF" and "KG" arrays, the base address + (rows * cols * 4) cannot exceed 65536. Keep in mind that the elements of the array are DWords, located on a DWord boundary. For example, AD0[4] will return AD0, AD4, AD8 and AD12.

For all arrays, the total number of bytes being requested cannot exceed the internal block size of 218 bytes.

Examples

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

2. To access Data Block 5 as word memory at byte offset 30, declare an address as follows:
DB5.DBW30

3. To access Data Block 2 as Boolean at byte offset 20 and bit 7, declare an address as follows:
DB2.DBX20.7

4. To access Data Block 1 as byte memory at byte offset 10, declare an address as follows:
DB1.DBB10

5. To access Flag 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

Data Block Strings

To reference Data Block Strings, the STRING subtype is used.

STRING Subtype

The STRING subtype follows the STEP 7 STRING data type definition. The syntax for the STRING subtype is *DBx.STRINGy.n* where *x* is the Data Block, *y* is the Byte offset, and *n* is the maximum String length. If *n* is not specified, the maximum String length will be 210 characters. String values read and written are stored at Byte offset *y+2* in Data Block *x*. The first two bytes contain the "maximum string length (*n*)" and the "actual string length". The "actual string length" gets updated with every write based on the string length of the string being written.

<i>y</i>	<i>y+1</i>	<i>y+2</i>	<i>y+3</i>	<i>y+4</i>	...	<i>y+2+n-1</i>
max string length (<i>n</i>)	actual string length	' '	' '	' '	...	' '

Note: STRING Strings are NULL padded. If the maximum string length is 10 and 3 characters are written, characters 4-10 are set to NULL.

Caution: When modifying Word, Short, DWord, Long and Float types, remember that each address starts at a byte offset within the device. Therefore, Words MW0 and MW1 overlap at byte 1. Writing to MW0 will also modify the value held in MW1. Similarly, DWord, Long and Float types can also overlap. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use MD0, MD4, MD8 and so on in order to prevent overlapping bytes.

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](#)

[Unable to write to '<address>' on device '<device name>'](#)

[Connection timeout on MPI node <device id>](#)

[Request timeout on MPI node <device id>](#)

Device Specific Messages

[Bad address in block '<start address>' to '<end address>' on device '<device name>.', the block has been deactivated](#)

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 statically has no length.

Solution:

Re-enter the address in the client application.

Device address '<address>' contains a syntax error

Error Type:

Warning

Possible Cause:

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

Solution:

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

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

Error Type:

Warning

Possible Cause:

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

Solution:

Modify the requested data type in the client application.

Device address '<address>' is Read Only

Error Type:

Warning

Possible Cause:

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

Solution:

Change the access mode in the client application.

Serial Communications

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

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

COMn does not exist

Error Type:

Fatal

Possible Cause:

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

Solution:

Verify that the proper COM port has been selected.

Error opening COMn

Error Type:

Fatal

Possible Cause:

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

Solution:

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

COMn is in use by another application

Error Type:

Fatal

Possible Cause:

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

Solution:

Verify that the correct port has been assigned to the channel.

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](#)[Unable to write to '<address>' on device '<device name>'](#)[Connection timeout on MPI node <device id>](#)[Request timeout on MPI node <device id>](#)

Device '<device name>' is not responding

Error Type:

Serious

Possible Cause:

1. The serial connection between the device and MPI adapter (or the MPI adapter 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 device and MPI adapter (or the MPI adapter and PC).
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 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.

Connection timeout on MPI node <device id>

Error Type:

Serious

Possible Cause:

1. The serial connection between the device and MPI adapter (or the MPI adapter 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 device and MPI adapter (or the MPI adapter and PC).
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.

Request timeout on MPI node <device id>

Error Type:

Serious

Possible Cause:

1. The serial connection between the device and MPI adapter (or the MPI adapter 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 device and MPI adapter (or the MPI adapter and PC).
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 Messages

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

Device Specific Messages

[Bad address in block '<start address>' to '<end address>' on device '<device name>.', the block has been deactivated](#)

Bad address in block '<start address>' to '<end address>' on device '<device name>.', the block has been deactivated

Error Type:

Serious

Possible Cause:

An attempt has been made to reference a block of memory that contains at least one nonexistent location in the specified device.

Solution:

Verify that the tags assigned to addresses are within the specified range on the device and eliminate ones that reference invalid locations.

Index

A

Address '<address>' is out of range for the specified device or register.....	10
Address Descriptions.....	7
Address Descriptions:Arrays.....	7
Address Validation.....	9

B

Bad address in block '<start address>' to '<end address>' on device '<device name>', the ..	13
block has been deactivated.....	
Boolean.....	6
Byte.....	6

C

Cable Connections.....	4
Communications error on '<channel name>' [<error mask>].....	11
COMn does not exist.....	10
COMn is in use by another application.....	11
Connection timeout on MPI node <device id>.....	12

D

Data Type '<type>' is not valid for device address '<address>'.....	10
Data Types Description.....	6
Device '<device name>' is not responding.....	12
Device address '<address>' contains a syntax error.....	9
Device address '<address>' is not supported by model '<model name>'.....	10
Device address '<address>' is read only.....	10
Device ID.....	4
Device Setup.....	4
Device Specific Messages.....	13
Device Status Messages.....	11

DWord..... 6

E

Error Descriptions..... 9

Error opening COMn..... 11

F

Float..... 6

L

Long..... 6

M

Master ID..... 4

Missing address..... 9

Modem Setup..... 5

N

Network..... 4

O

Overview..... 3

R

Request timeout on MPI node <device id>..... 12

S

Serial Communications..... 10
Short..... 6
String..... 6

U

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

W

Word..... 6