Philips P8/PC20 Driver Help

Table of Contents

Philips P8/PC20 Driver Help

Help version 1.014

CONTENTS

Overview

The Philips P8/PC20 Driver provides an easy and reliable way to connect Philips P8/PC20 devices to OPC Client applications, including HMI, SCADA, Historian, MES, ERP and countless custom applications. It is intended for use with Philips P8 and PC20 devices.

Note: A Philips P8 device is byte (8 bits) oriented and will always use an offset in words even when the requested data type is smaller. A Philips PC20 device is nibble (4 bit) oriented. It will only use the least significant nibble of a byte to store its data.

Device Setup

Supported Devices

PC20*
MC30**
MC31
MC41
ISCOS 10
ISCOS 60
ISCOS 70

*Using the CI21 or VI21 card.

**Using the CI30 card.

Communication Protocol

Philips Programmable Controller Communication Protocol (PPCCOM)

Supported Communication Parameters

Baud Rate: 9600, 19200
Parity: Even
Data Bits: 8
Stop Bits: 1 or 2

Note: Not all devices support the configurations listed above.

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. It may be invoked through the COM ID dialog in Channel Properties. For more information, refer to the OPC server's help documentation.

Device IDs

The Device ID (which is the same as a slave number) is a unique number used to identify a PLC. Each PLC must have its own unique number. The Device ID is used in the protocol so only the PLC with the requested slave number will reply. It can range from 0 to 15. Since there are 16 available Device IDs, a network can hold a maximum of 16 PLCs.

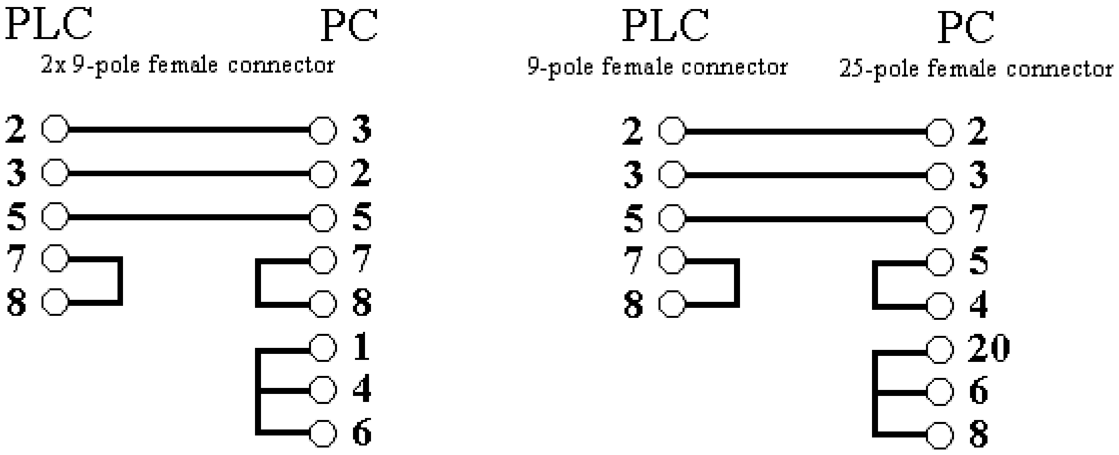
Flow Control

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

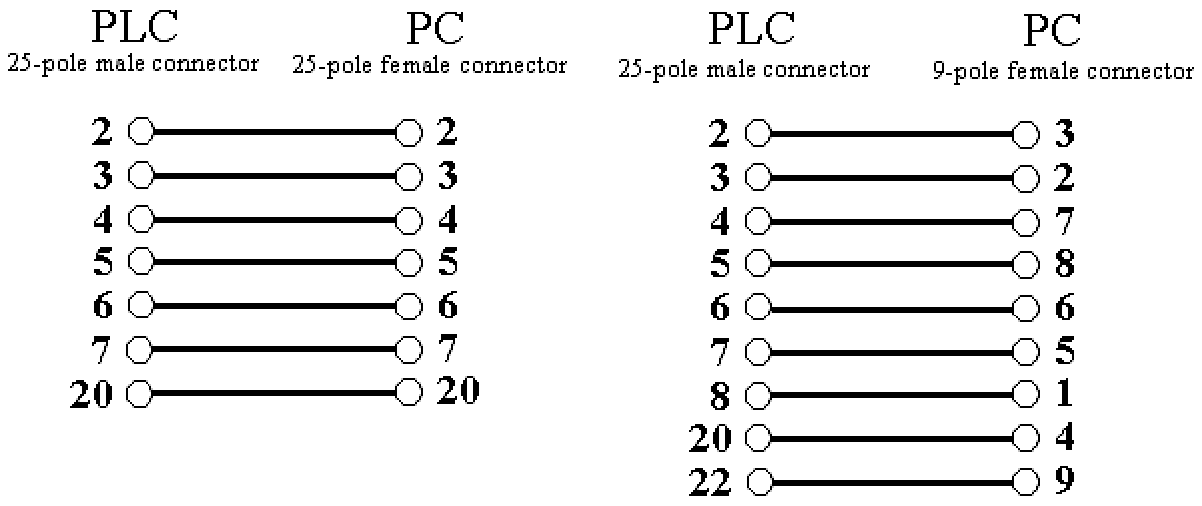
Note: When using the manufacturer's supplied communications cable, it is sometimes necessary to choose a flow control setting of **RTS** or **RTS Always** under the Channel Properties.

Cable Connections

Cable Connections for a P8 Model Device



Cable Connections for a PC20 Model Device



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	PC20	P8
Boolean	Single bit	Supported	Supported
Byte	Unsigned 8 bit value bit 0 is the low bit bit 7 is the high bit	Supported	Supported
Char	Signed 8 bit value bit 0 is the low bit bit 6 is the high bit bit 7 is the sign bit	Supported	Supported
Word	Unsigned 16 bit value byte 0 is the high byte byte 1 is the low byte	Supported	Supported
Short	Signed 16 bit value byte 0 is the high byte byte 1 is the low byte bit 0 is the sign bit	Not Supported	Supported
DWord	Unsigned 32 bit value byte 0 is the high byte byte 7 is the low byte	Supported	Supported
Long	Signed 32 bit value byte 0 is the high byte byte 7 is the low byte bit 0 is the sign bit	Not Supported	Supported
Double	64 bit floating point value	Supported	Not Supported

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.

PC20 Addressing
P8 Addressing

PC20 Addressing

The syntax for PC20 device addressing is not IEC 1131 compliant. The PPCCOM PLC memory architecture is based on nibbles, which are 4 bit units capable of representing decimal values ranging from 0 to 15. Even if a discrete (bit)-type item is defined, the actual communication between the PPCCOM Driver and the PLC handles nibbles. The general addressing syntax is as follows:

PAAADN(N)

Descriptions of the components are as follows:

- *P* is the RAM page number. It can range from 0 to 3.
- *AAA* is the address, which must always be three characters wide. It can range from 000 to 511 (inclusive).
- *D* indicates the data type.
  - A period ('.') should be used for discrete (Boolean) data.
  - A plus ('+') should be used for 'nibble in integer.'
  - A colon (':') should be used for the nibble string data format.
- *N(N)* is a number. It holds a bit position between 0 and 3 in case of the discrete data type (or a length in case of the two other data types). When using 'nibble in integer' the data type length must be between 1 and 8. The length in case of the nibble string data format is practically limited to the time that the PLC needs to calculate the CRC checksum.

Examples

1. 0001+2
Word data type starting at nibble 1 in RAM page 0.

2. 2444.3
Bit 3 from nibble 444 in RAM page 2.

3. 3245:21
21 nibbles with their hexadecimal values represented as characters starting at nibble 245 in RAM page 3.

4. 1102+7
Double data type starting at nibble 102 in RAM page 1.

P8 Addressing

The syntax for P8 device addressing is IEC 1131 compliant. The default data types for dynamically defined tags are shown in **bold**.

Device Type	Data Range	Data Type	Access
Discrete Inputs	%IX0.b-%IX1023.b b is bit number 0-15.	**Boolean**	Read Only
Discrete Inputs	%IB0.b-%IB1023.b b is byte number 0-1.	**Byte**, Word, DWord, Double	Read/Write
Discrete Inputs	%IW0-%IW1023	**Word**, DWord, Double	Read/Write
Discrete Inputs	%ID0-%ID1023	**DWord**, Double	Read/Write
Discrete Outputs	%QX0.b-%Qx1023.b b is bit number 0-15.	**Boolean**	Read Only
Discrete Outputs	%QB0.b-%QB1023.b b is byte number 0-1.	**Byte**, Word, DWord, Double	Read/Write
Discrete Outputs	%QW0-%QW1023	**Word**, DWord, Double	Read/Write

Discrete Outputs	%QD0-%QD1023	**DWord**, Double	Read/Write
Memory	%MX0.b-%MX14335.b b is bit number 0-15.	**Boolean**	Read Only
Memory	%MB0.b-%MB14335.b b is byte number 0-1.	**Byte**, Word, DWord, Double	Read/Write
Memory	%MW0-%MW14335	**Word**, DWord, Double	Read/Write
Memory	%MD0-%MD14335	**DWord**, Double	Read/Write

The actual number of addresses for each type depends on the P8 type device in use. Each type does not necessarily support an address of 0 to 1023 or 0 to 14335. For address ranges, refer to the device documentation.

Note 1: In case of bit access, the bit index differs depending on the device type accessed. When reading from an Input or Output, the Least Significant Bit (LSB) will be bit 0 and the Most Significant Bit (MSB) will be bit 15. When reading from or writing to memory, the Least Significant Bit (LSB) will be bit 15 and the Most Significant Bit (MSB) will be bit 0.

Note 2: In case of byte access, the byte index will be 0 for the Most Significant Byte and 1 for the Least Significant Byte.

Examples

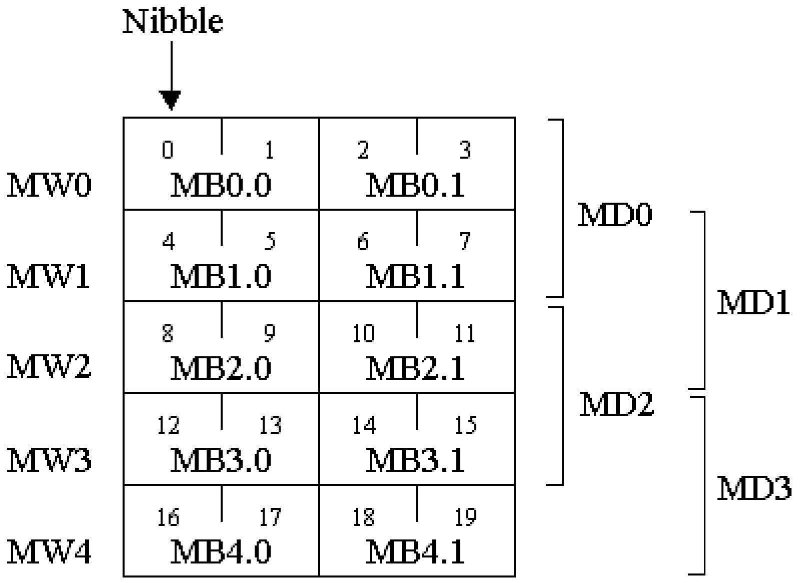
1. %MW10
Word 10 in memory area.

2. %ID123
Double word 123 in input area.

3. %MB5.0
Most significant byte in input word 5.

4. %QX0.6
Bit 6 in output word 0.

Caution: When modifying DWord types, remember that each address starts at a word offset within the device. Therefore, DWords MD0 and MD1 overlap at word 1. Writing to MD0 will also modify the value held in MD1. It is recommended that these memory types be used so that overlapping does not occur. For example, when using DWords, use MD0, MD2, MD4 ... and so on in order to prevent overlapping words. For more information, refer to the diagram below.



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>'

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 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.

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.

Unable to set comm parameters on COMn

Error Type:
Fatal

Possible Cause:
The serial parameters for the specified COM port are not valid.

Solution:
Verify the serial parameters and make any necessary changes.

Communications error on '<channel name>' [<error mask>]

Error Type:
Serious

Error Mask Definitions:
B = Hardware break detected.
F = Framing error.
E = I/O error.
O = Character buffer overrun.
R = RX buffer overrun.
P = Received byte parity error.
T = TX buffer full.

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

Solution:
1. Verify the cabling between the PC and the PLC device.
2. Verify 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.
4. The response from the device took longer to receive than the amount of time specified in the "Request Timeout" device setting.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify the specified communications parameters match those of the device.
3. Verify that the Network ID given to the named device matches that of the actual device.
4. Increase the Request Timeout setting so that the entire response can be handled.

Unable to write to '<address>' on device '<device name>'

Error Type:

Serious

Possible Cause:

1. The serial connection between the device and the Host PC is broken.
2. The communications parameters for the serial connection are incorrect.
3. The named device may have been assigned an incorrect Network ID.

Solution:

1. Verify the cabling between the PC and the PLC device.
2. Verify 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.

Index