

## Tips using the EA-1 with a PLC

By Michael Doherty, Applications Engineering Manager

*The Ophir EA-1 Adapter is primarily for OEM customers who wish to embed the device inside their own laser system.*

In order to allow customers to control the device using their own software package or PLC, a set of "User Commands" are provided which allow control and query of all the features of the device. For "User Command" details please see the EA-1 user manual.

### TCP Connection and ports:

The EA-1 can communicate via Ethernet using three different connection types each using their own specific ports. In this document we will only review Telnet and UDP connection types as these would be the most widely used for PLC integration.

Connection:	Port Number:
Telnet	23
HTTP	80
UDP	11000

### UDP Protocol:

The UDP protocol is the easiest to set up but this connection type has its drawbacks. UDP is a broadcast protocol so all devices on the network receive the broadcast. There is no confirmation that the packet has been received by a device or if a duplicate packet has been received.

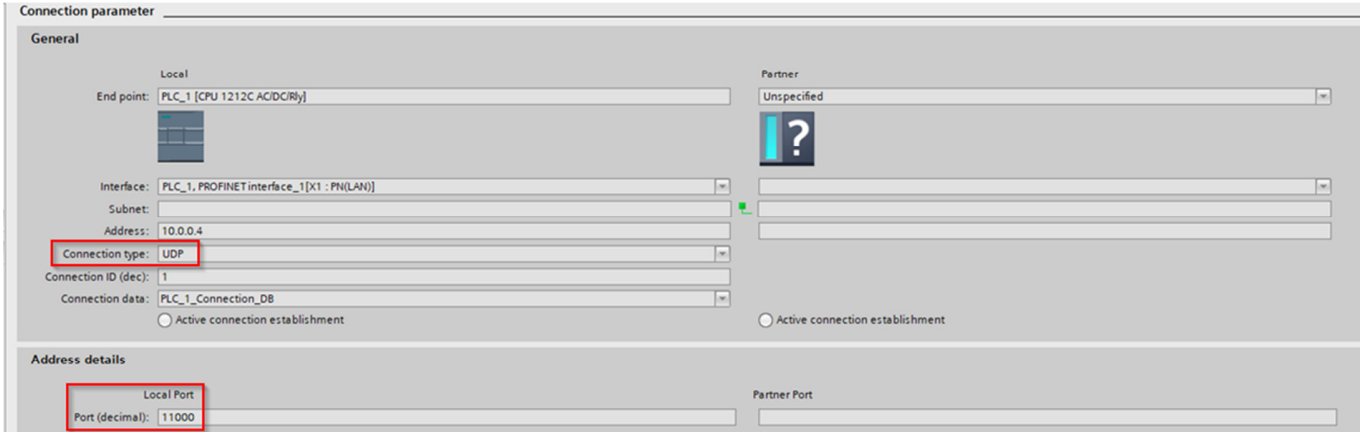
Siemens PLC Configuration: (Figure 1)

PLC is the Local Device

EA-1 is the Partner Device

Given that UDP is a broadcast protocol. The partner device info does not have to be defined in the connection parameters. The only things needing definition are the circled items. The Connection Type: UDP and Local Port 11000.

Figure 1



### UDP PLC Communication Hints:

Siemens - see commands **TUSEND** and **TURCV**

Beckhoff - see commands **ITcIoUdpProtocol** and **ITcIoUdpProtocolRecv**

### Example Communication using Siemens PLC:

#### Send Data: (Figure 2)

1. The first section (in blue) specifies the intended IP address for the command, reading down the IP subnets are 10-0-0-2. This corresponds to the IP address of the EA-1. In this case the IP address is: 10.0.0.2
2. The second section (in purple) specifies the UDP port that the command will be sent on.
3. The third section (in red) specifies the command that will be sent to the EA-1. Reading down "OPHCMD1211\$SP\$R in this case the \$R stands for [CR] thus ending the command. For details of this UDP command see the EA-1 Manual.

Figure 2

"TUSEND_address".Parametri	DEC	10		
"TUSEND_address".Parametri	DEC	0		
"TUSEND_address".Parametri	DEC	0		
"TUSEND_address".Parametri	DEC	2		2
"TUSEND_address".Parametri	DEC	11000		
"TUSEND_data".string[1]	Character	'O'		'O'
"TUSEND_data".string[2]	Character	'P'		'P'
"TUSEND_data".string[3]	Character	'H'		'H'
"TUSEND_data".string[4]	Character	'C'		'C'
"TUSEND_data".string[5]	Character	'M'		'M'
"TUSEND_data".string[6]	Character	'D'		'D'
"TUSEND_data".string[7]	Character	'1'		'1'
"TUSEND_data".string[8]	Character	'2'		'2'
"TUSEND_data".string[9]	Character	'1'		'1'
"TUSEND_data".string[10]	Character	'1'		'1'
"TUSEND_data".string[11]	Character	'\$\$'		'\$\$'
"TUSEND_data".string[12]	Character	'S'		'S'
"TUSEND_data".string[13]	Character	'P'		'P'
"TUSEND_data".string[14]	Character	'\$R'		'\$R'

**Receive Data:** (Figure 3)

Looking at the receive array of the PLC after sending the command above. Reading from the top down

"OPHRSP1211\*0.09E-3\$R\$L"

For details on reading and evaluating the UDP response, please see the EA-1 Manual.

Figure 3

"TURCV_data".Data[1]		Character	'O'
"TURCV_data".Data[2]		Character	'P'
"TURCV_data".Data[3]		Character	'H'
"TURCV_data".Data[4]		Character	'R'
"TURCV_data".Data[5]		Character	'S'
"TURCV_data".Data[6]		Character	'P'
"TURCV_data".Data[7]		Character	'1'
"TURCV_data".Data[8]		Character	'2'
"TURCV_data".Data[9]		Character	'1'
"TURCV_data".Data[10]		Character	'1'
"TURCV_data".Data[11]		Character	'*'
"TURCV_data".Data[12]		Character	'0'
"TURCV_data".Data[13]		Character	'.'
"TURCV_data".Data[14]		Character	'0'
"TURCV_data".Data[16]		Character	'9'
"TURCV_data".Data[17]		Character	'E'
"TURCV_data".Data[18]		Character	'.'
"TURCV_data".Data[19]		Character	'3'
"TURCV_data".Data[20]		Character	'\$R'
"TURCV_data".Data[21]		Character	'\$L'

### Telnet Protocol:

Telnet is a Client-Server protocol with reliable packet confirmation packets. The EA-1 is defined as the Telnet server and the PLC or other Ethernet device is the client. Communication will be established over port 23. Since Telnet is a defined protocol that usually isn't supported directly by a PLC, it is necessary to emulate the IAC bytes (Interpret as Command) of the protocol. Some prefix values for the Telnet protocol will need to be sent prior to the EA-1 command.

Siemens PLC Configuration: (Figure 4)

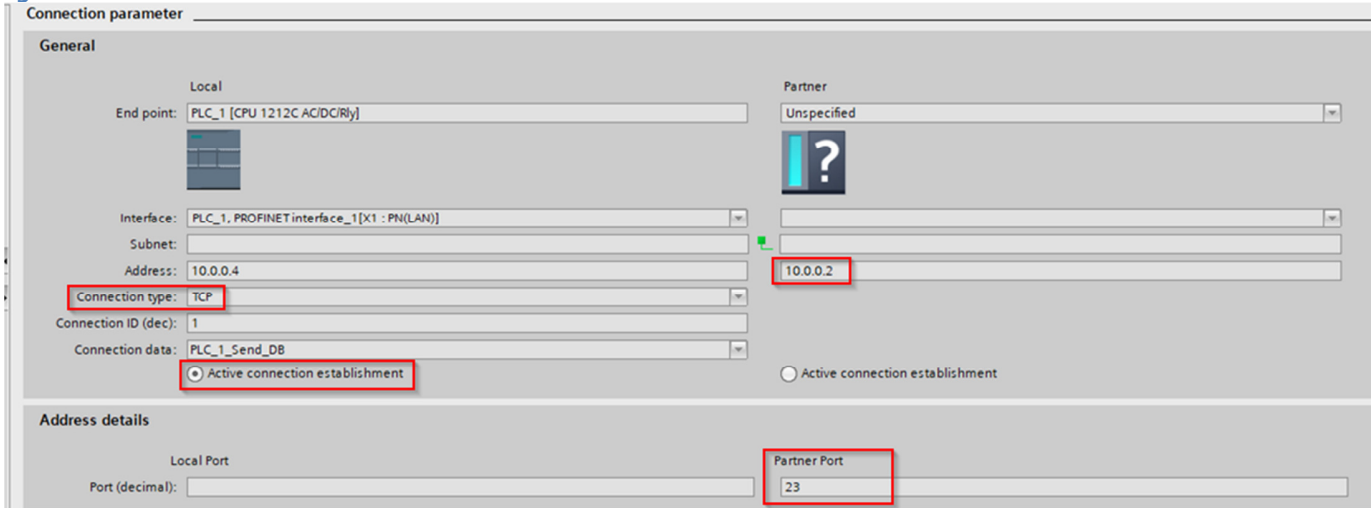
PLC is the Local Device

EA-1 is the Partner Device

With TCP/IP the connection has to be established to a specific IP address and port. Figure 4 illustrates the connection parameters.

- Partner IP address is set to that of the EA-1 (10.0.0.2) and the port is set to 23.
- The PLC connection type is set to TCP.
- Selection of the device that will establish communication needs to be also picked. In this case the PLC will always request data from the EA-1 so select the PLC as the "Active Connection establishment".

Figure 4



TCP/IP PLC Communication Hints:

Siemens - see commands similar to **TCONN**, **TSEND** and **TRCV**

Beckhoff - see commands **ITcIoTcpProtocol** and **ITcIoTcpProtocolRecv**

Example Communication using Siemens PLC:

**Send Data:** (Figure 5)

1. The first section (in blue) specifies the IAC commands for emulating the Telnet protocol. The IAC command is 0xFF 0xFE 0x18 0xFF 0xF0
2. The second section (in purple) specifies the command that will be sent to the EA-1. Reading down "\$SP\$R\$L in this case the \$R stands for [CR] and \$L stands for (LF) for details of the \$SP (Send Power) command see the EA-1 Manual.

Figure 5

"Data_block_1"."Send_data(DATA)"[0]	Hex	16#FF	16#FF
"Data_block_1"."Send_data(DATA)"[1]	Hex	16#FE	16#FE
"Data_block_1"."Send_data(DATA)"[2]	Hex	16#18	16#18
"Data_block_1"."Send_data(DATA)"[3]	Hex	16#FF	16#FF
"Data_block_1"."Send_data(DATA)"[4]	Hex	16#F0	16#F0
"Data_block_1"."Send_data(DATA)"[5]	Character	'\$\$'	'\$\$'
"Data_block_1"."Send_data(DATA)"[6]	Character	'S'	'S'
"Data_block_1"."Send_data(DATA)"[7]	Character	'P'	'P'
"Data_block_1"."Send_data(DATA)"[8]	Character	'\$R'	'\$R'
"Data_block_1"."Send_data(DATA)"[9]	Character	'\$L'	'\$L'

**Receive Data:** (Figure 6)

Looking at the receive array of the PLC we can break this into sections.

1. The first section (in blue) the EA-1 Echo's the command that was sent.
2. The second section (in purple) we get the values for the command that was sent.
3. Reading from the top down we get \*0.019E-3\$R\$L> again for details of this response please see the EA-1 manual.

Figure 6

*Data_block_2*.Receive_data[1]	Character	'\$\$'
*Data_block_2*.Receive_data[2]	Character	'S'
*Data_block_2*.Receive_data[3]	Character	'P'
*Data_block_2*.Receive_data[4]	Character	'\$R'
*Data_block_2*.Receive_data[5]	Character	'\$L'
*Data_block_2*.Receive_data[6]	Character	'*'
*Data_block_2*.Receive_data[7]	Character	'0'
*Data_block_2*.Receive_data[8]	Character	'.'
*Data_block_2*.Receive_data[9]	Character	'0'
*Data_block_2*.Receive_data[10]	Character	'1'
*Data_block_2*.Receive_data[11]	Character	'9'
*Data_block_2*.Receive_data[12]	Character	'E'
*Data_block_2*.Receive_data[13]	Character	'.'
*Data_block_2*.Receive_data[14]	Character	'3'
*Data_block_2*.Receive_data[15]	Character	'\$R'
*Data_block_2*.Receive_data[16]	Character	'\$L'
*Data_block_2*.Receive_data[17]	Character	'>'

### Closing Socket or Telnet connection:

The EA-1 will send 6 bytes of data upon terminating the socket connection. For reference the following data will be sent by the EA-1 upon termination of the connection (Figure 7).

0xFF 0xFD 0x24 0xFF 0xFB 0x01

Figure 7

*Data_block_2*.Receive_data[1]	Hex	16#FF
*Data_block_2*.Receive_data[2]	Hex	16#FD
*Data_block_2*.Receive_data[3]	Hex	16#24
*Data_block_2*.Receive_data[4]	Hex	16#FF
*Data_block_2*.Receive_data[5]	Hex	16#FB
*Data_block_2*.Receive_data[6]	Hex	16#01