

EPower™ Controller Communications Manual

User Guide

HA179770 Issue 8

Date: 09/2020

Eurotherm®

by Schneider Electric

Table of Contents

- Table of Contents 3
- Application of User Communications Timeout 7
 - Fieldbus Status Parameter..... 7
 - Parity Parameter 8
 - Modbus and Modbus/TCP Timeout..... 8
 - Usage..... 9
- Introduction 11
 - Jbus V Modbus 11
 - References..... 12
 - EIA232, EIA422 And EIA485 Transmission Standards 12
 - Cable Selection 14
 - Wiring General 15
 - Precautions 16
 - Grounding 16
- Digital Communications Hardware 17
 - Communications Pinouts 17
 - Modbus RTU 17
 - Modbus TCP (Ethernet 10baseT), Single Port..... 17
 - Modbus TCP (Dual Port)..... 18
 - DeviceNet 18
 - Profibus 19
 - EtherNet/IP (Single Port) 19
 - EtherNet/IP (Dual Port) 20
 - CC-Link 20
 - PROFINET IO (Single Port) 21
 - PROFINET IO (Dual Port)..... 22
 - Wiring 3-Wire EIA485..... 22
 - Interconnection Diagram 3-Wire EIA485 23
 - iTools 25
 - Modbus Parameters 26
 - Comms Identity 26
 - Protocol 26
 - Unit Address..... 26
 - Baud Rate 26
 - Parity..... 27
 - Delay..... 27
- Modbus and Jbus Protocol 29
 - Protocol Basics 29
 - Typical Transmission Line Activity..... 30
 - Device Address 30
 - Parameter Address 30
 - Parameter Resolution 30
 - Reading of Large Numbers 31
 - Mode of Transmission..... 31
 - Message Frame Format..... 31
 - Cyclic Redundancy Check 32
 - Example of a CRC Calculation..... 33
 - Example of a CRC Calculation in the 'C' Language..... 35
 - Example of a CRC Calculation in Basic Language 35
 - Function Codes 35
 - Read N Words..... 36
 - Write A Word 37
 - Diagnostic Loopback..... 37
 - Write N Words 38

- Error Response 39
- Wait Period..... 40
- Latency..... 40
- Message Transmission Time..... 40
- Status Words..... 41
- Fieldbus Comms Status 41
- Strategy Status Word 42
- Error Status Words..... 42
- Status Word to Indicate Instrument Errors Via Comms 42
- Configuration Mode Parameters 42
- Modbus Advanced Topics 45**
 - Access to Full Resolution Floating Point and Timing Data..... 45
 - Data Types Used in EPower Controller Units 45
 - Enumerated, Status Word, and Integer Parameters 46
 - Floating Point Parameters..... 46
 - Time Type Parameters 47
 - Programmable Logic Controllers and EPower Controller 48
- Ethernet (Modbus TCP), Single Port 49**
 - Overview 49
 - Support for other Ethernet Utilities..... 49
 - Ethernet Wiring 49
 - Instrument Setup..... 50
 - Unit Identity 51
 - Dynamic Host Configuration Protocol (DHCP) Settings 51
 - Fixed IP Addressing 51
 - Dynamic IP Addressing..... 51
 - Default Gateway 51
 - Preferred Master 52
 - Itools Setup 52
- Ethernet (Modbus TCP), Dual Port 55**
 - Overview 55
 - Function Codes and Address Space..... 55
 - Support for other Ethernet Utilities..... 56
 - Ethernet Wiring 56
 - Instrument Setup..... 57
 - Itools Setup 57
- Profibus 59**
 - Introduction 59
 - Wiring General 60
 - Cable Selection 61
 - Maximum Baud Rate Compared with Cable Length 61
 - Node Address..... 61
 - Configuring the Data Exchange 62
 - To Configure the Profibus Master..... 66
 - DPV1 Acyclic Communications 67
 - Trouble-Shooting..... 67
- DeviceNet 69**
 - Introduction 69
 - EPower Controller DeviceNet Features 69
 - DeviceNet Wiring 70
 - Setting Up the EPower Controller Unit..... 71
 - Unit Address..... 71
 - Baud Rate 72
 - Data Exchange Mapping..... 72
 - Configuring the Data Exchange 72
 - Setting up the Master 75
 - Establishing Communications 75
 - Data Formats 75

Explicit Messaging	75
The EDS File	76
Trouble-Shooting	76
Ethernet/IP™	77
Introduction	77
EPower Controller EtherNet/IP Features	77
Ethernet/IP Wiring	78
Setting Up the EPower Controller Unit	78
Dynamic Host Configuration Protocol (DHCP) Settings	79
Fixed IP Addressing	79
Dynamic IP Addressing	79
Default Gateway	79
Data Exchange Mapping	80
Configuring the Cyclic (Implicit) Data Exchange	81
Setting Up the Master	83
Cyclic (Implicit) Data Exchange	83
Acyclic (Explicit) Messaging	84
Establishing Communications	84
Data Formats	85
The EDS File	85
Troubleshooting	85
CC-Link	87
Introduction	87
EPower Controller CC-Link Features	87
CC-Link Wiring	88
Maximum Transmission Distance	89
Setting Up the EPower Controller Unit	89
Unit Address (CC-Link Station Number)	89
Baud Rate	89
Occupied Stations	90
Data Exchange Mapping	90
Configuring the Data Exchange	91
Setting Up the Master	94
Examples	94
CC-Link System Area	94
System Area Layout	94
System Area Location	95
System Area Flag Handshaking	96
Establishing Communications	96
Data Formats	96
Troubleshooting	97
PROFINET	99
Introduction	99
EPower Controller PROFINET Features	99
PROFINET Wiring	99
Setting Up the EPower Controller Unit	100
PROFINET Initialisation Mode (PninitMode) Parameter	100
Dynamic Host Configuration Protocol (DHCP) Settings	101
Fixed IP Addressing	101
Dynamic IP Addressing	101
Default Gateway	101
Device Name	103
Data Exchange Mapping	104
Configuring the Cyclic (Implicit) Data Exchange	105
Acyclic (Explicit) Messaging	107
PROFINET Acyclic Readings	107
Constraints on the Parameters	108
Data Formats	108
The GSD File	109
Example - Using A PLC to Configure Epower as a PROFINET I/O Device ...	109

Requirements.....	109
Solution overview.....	109
Information about the Ethernet Configuration.....	110
PLC Configuration.....	110
Insert a PLC into the Project.....	112
Add a Rail, the Power Module, the PLC and the PROFINET Module.	113
STEP7 First-time Configuration – Install the GSD File	115
Add the EPower Device to the Configuration.....	118
Configure the IP Address and the Device Name	118
To Configure the Application	121
I/O Configuration.....	122
Troubleshooting	124
References.....	124
Appendix A - Warning	125
Continuous Writing to Parameters	125
Solution:	127
Scaled Integers	128
Re-scaling	128
Parameters Which Always Require Rescaling.....	128
Conditional Re-scaling	129
Energy Counter Scaling.....	131
Appendix B Communication Enhancement Modbus TCP and Modbus RTU	133
Configuring the Block Read and Write Table	133
Appendix C. Glossary of Terms	137
Appendix D. ASCII Codes	139

Issue Status Of This Manual

Issue 1 applies to EPower controller units fitted with firmware version 2.00.

Issue 2 includes CC-Link and EtherNet I/P communications. Appendix A now provides a warning concerning continuous writes to EEPROM and scaling of integers.

Issue 3 includes a new chapter (10) for PROFINET communications and Appendix B Modbus TCP and Modbus RTU enhancements.

Issue 4 includes updates to the Profinet chapter 10.

Issue 5 updates paragraphs 7.4 - Devicenet and 8.4 - Ethernet - to 32 input and 16 output variables.

Issue 6 adds features available from firmware version 3.05 and above. They are described in Chapter 1 inserted at the beginning of the manual and updates to Continuous Writing to Parameters in Appendix A.

Issue 7 adds features now available in and supported by firmware version 3.10 and above. The feature consists of three different communication module types; ModbusTCP dual port, EtherNet/IP dual port and Profinet IO dual port (see "Digital Communications Hardware").

Application of User Communications Timeout

This chapter provides an update to issue 5 of the EPower Digital Communications Handbook and includes features and additional parameters introduced in firmware versions V3.05 and above. It applies to ALL EPower communications protocols.

Fieldbus Status Parameter

This section applies to the following communications protocols:

ModbusTCP (dual port), Profibus, Devicenet, CanOpen, CCLink, Profinet IO (single and dual port) and Ethernet/IP (single and dual port).

It does not apply to Modbus and Modbus/TCP single port.

The Fieldbus Status parameter is found in the 'Faultdet.CommsStatus' list. This parameter is a raw copy of the 'NetStatus' parameter found in the 'Comms.User.NetStatus' list and a value of '4' (Active) means that communications with the master is running correctly. However, in firmware versions 3.04 and below it is not possible to wire 'Faultdet.CommsStatus' to the setpoint selector parameter ('SPselect' or 'RemSelect') in setpoint provider directly.

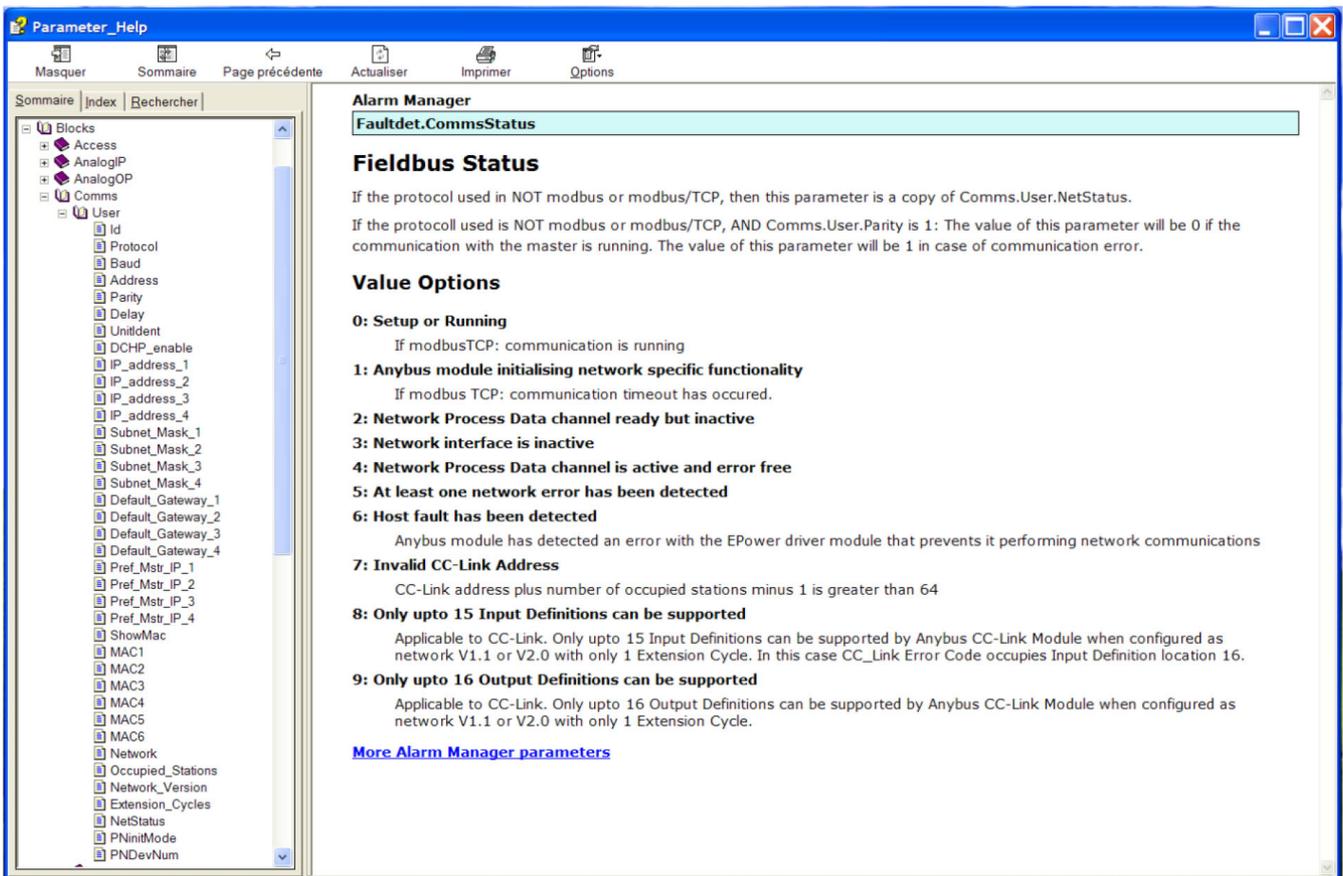


Figure 1 Fieldbus Status Parameter

Parity Parameter

From V3.05, this behaviour is the same by default, but a new mode has been added as follows:

If the parameter 'Comms.User.Parity' is set to '1' (Even), then the 'FaultDet.CommsStatus' parameter will only take values between '0' and '1' - 0 if communications is running (if 'Comms.User.NetStatus' = Active) or 1 otherwise (communications is not running).

This allows the 'FaultDet.CommsStatus' parameter to be wired directly to the setpoint selector parameter ('SPselect' or 'RemSelect') without using any more function blocks.

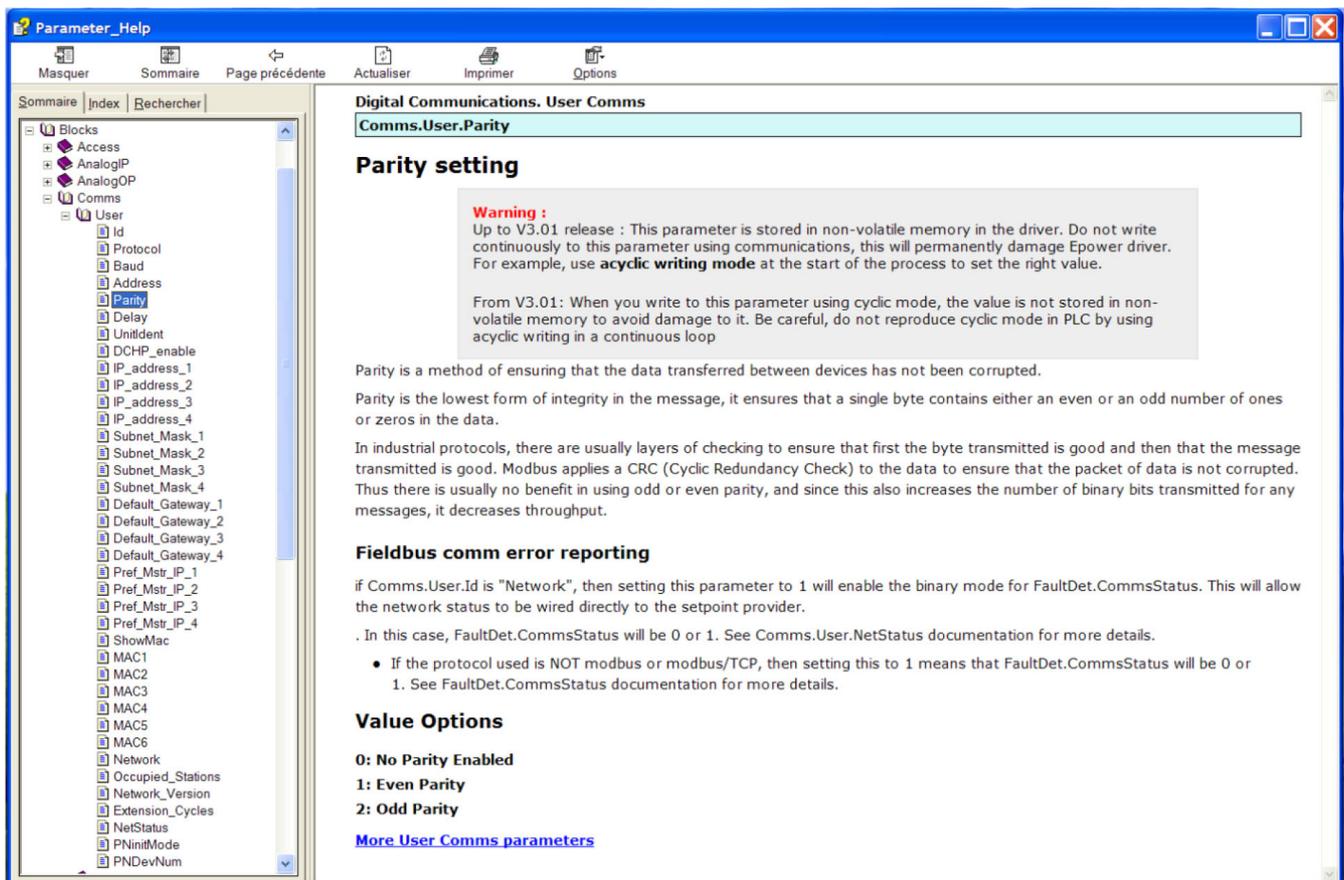


Figure 2 Parity Parameter

Modbus and Modbus/TCP Timeout

This section applies to communications protocols Modbus and Modbus/TCP single port only.

From firmware versions V3.05 and above, a new feature has been added. This is timeout of communications for Modbus and Modbus/TCP single port. The timeout parameter is found as 'Comms.User.PNDevNum'.

To activate this feature, set a timeout value in 'Comms.User.PNDevNum'. A value of '0' disables this facility.

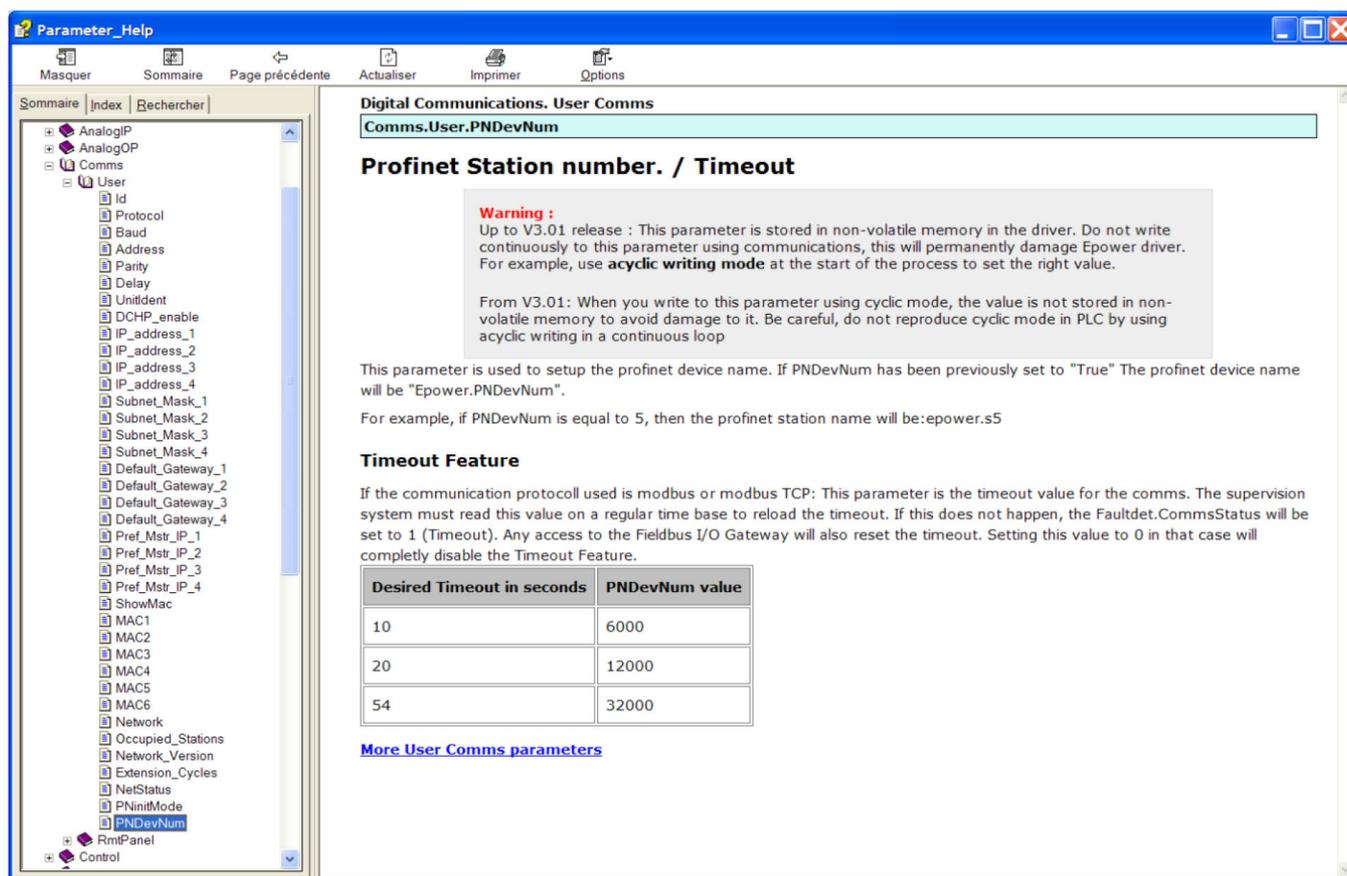


Figure 3 Timeout Parameter

Timeout operates as follows:

A counter is made to operate:

- If there is an IO/gateway access. The counter is reloaded with the timeout value entered in 'Comms.User.PNDevNum'. 'Faultdet.CommsStatus' is set back to 0.
- If there is an acyclic modbus read request on the parameter 'Comms.User.PNDevNum'. The counter is then reloaded with the timeout value from 'Comms.User.PNDevNum'. 'Faultdet.CommsStatus' is set back to 0.

As the counter reaches zero, 'Faultdet.CommsStatus' is set to 1.

The desired timeout is set according to the table below:

Desired Timeout in seconds	PNDevNum value
10	6000
20	12000
30	32000

Usage

'Faultdet.CommsStatus' can be wired directly to the 'SetProv.SPSelect' of the setpoint provider block which allows a fallback value to be used in case of loss of communication.

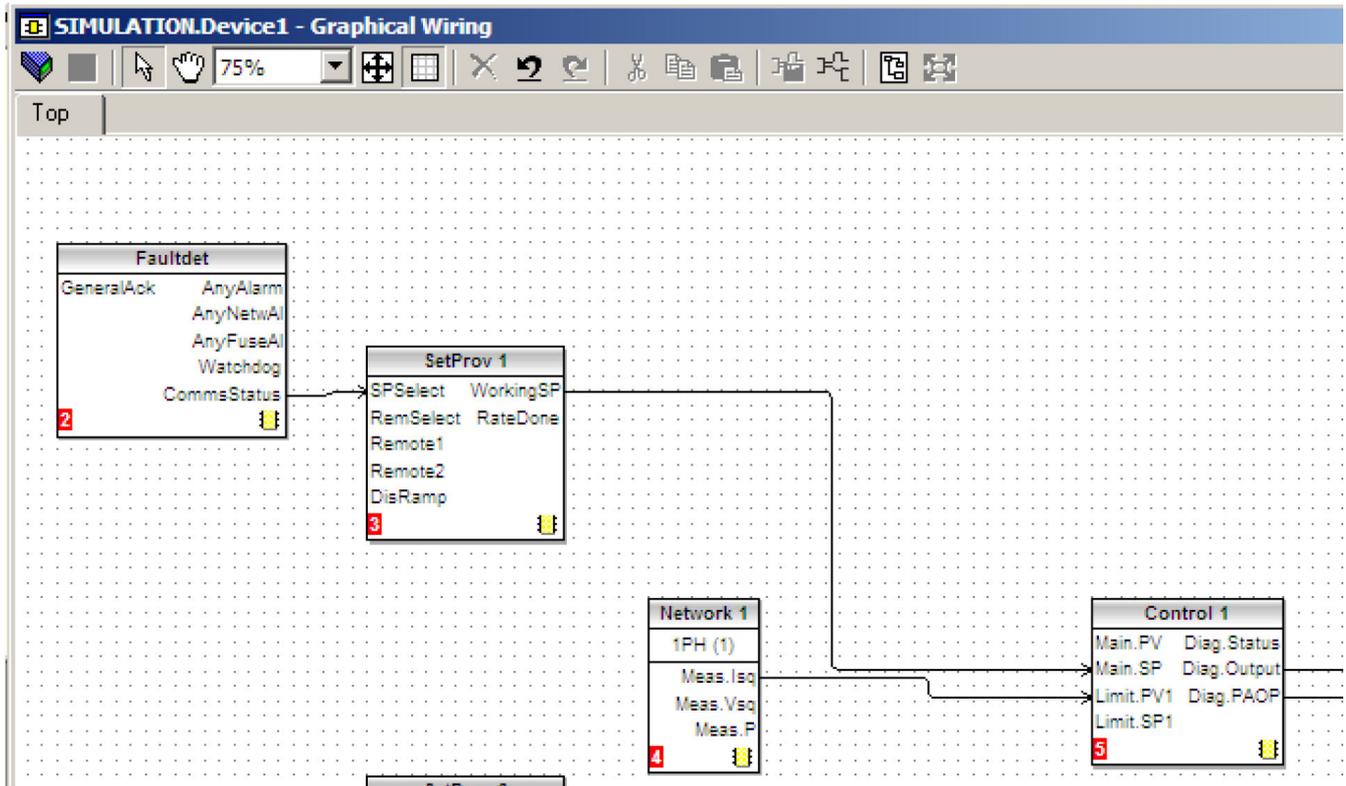


Figure 4 Example of wiring using Modbus Timeout Parameter

Introduction

This handbook is written for those people who need to use a digital communications link and Modbus, (Jbus), DeviceNet®, Profibus, Modbus TCP (Ethernet), EtherNet/IP™ or CC-Link communication protocols to supervise EPower controller power management and control units.

Chapter 1 describes parameter additions in firmware version 3.05

Chapter 2 provides general information on standards, cabling and electrical connections.

Chapter 3 gives connector and wiring details for a serial link

Chapter 4 describes Modbus RTU and Jbus protocols.

Chapter 5 covers advanced topics such as access to full resolution floating point data and user interface permissions.

Chapter 6 describes Ethernet (Modbus TCP)

Chapter 8 describes Profibus

Chapter 9 describes DeviceNet®

Chapter 10 describes EtherNet/IP™

Chapter 11 describes CC-Link

Chapter 12 describes PROFINET

Appendix A Warning - Continuous writing to parameters and Integer Scaling

Appendix B is a Glossary of Terms.

Appendix C lists ASCII codes.

Jbus V Modbus

- Modbus is a serial communications protocol and is a registered trademark of Gould Modicon Inc.
- Jbus differs from Modbus in that the parameter addresses are displaced by 1.
- The two protocols use the same message frame format.
- There is a total of 16 Modbus and Jbus function codes but EPower controller units use a subset - that is 3, 4, 6 and 16.
- In this document reference will be made to Modbus, however all information applies equally to Jbus.

References

Refer to the documents and links below for further information.

Gould	Modbus Protocol Reference Guide, PI-MBUS-300	
April	Jbus Specification	
EIA Standard EIA232-C	Interface Between Terminal Equipment and Data Communication Equipment Employing Serial Binary Interchange	EIA232 is not supported in the current version of EPower controller
EIA Standard EIA485	Electrical Characteristics of Generators and Receivers for use in Balanced Digital Multipoint Systems	In this manual referred to as EIA485
EIA Standard EIA422	Electrical Characteristics of Balanced Voltage Digital Interface Circuits	EIA422 is not supported in the current version of EPower controller
HA179769	The User Guide describes how to install, wire, use and configure the units. It also lists Modbus parameter addresses.	Can be downloaded from www.eurotherm.com
HA026230	Series 2000 Communication Handbook	Can be downloaded from www.eurotherm.co.uk
KD485	KD485 Installation and Operation Handbook	Supplied with the KD485 unit
iTools	This is a software package for configuring Eurotherm instruments. EPower controller units may be configured using iTools from version 7.00.	
EtherNet/IP	ODVA Technologies: EtherNet/IP Library	www.odva.org .
CC-Link	CC-Link Partner Association	www.cc-link.org .
DeviceNet	ODVA Technologies: DeviceNet Library	www.odva.org .
Profibus	Profibus DP	www.profibus.com .

EIA232, EIA422 And EIA485 Transmission Standards

Serial communications for Modbus RTU, is performed using EIA485 only in the current version EPower controller units. However, other instruments in the Eurotherm range support other standards and it is appreciated that it may be necessary to use EPower controller units in existing installations using other standards. These are discussed below.

The Electrical Industries Association, (EIA) introduced the Recommended Standards, EIA232, EIA485 and EIA422 (formerly RS232, RS485 and RS422). These standards define the electrical performance of a communications network. The table below is a summary of the different physical link offered by the three standards.

EIA Standard	EIA232C	EIA485 3-Wire	EIA485 5-Wire (EIA422)
Transmission mode	Single ended +12volts nominal (3v minimum 15V maximum)	One or Two Pairs of wires. Differential Mode. Half duplex - communication occurs in both directions but not at the same time. Typically once a unit begins receiving a signal it must wait for the transmitter to stop sending before it can reply.	Differential Mode using two pairs of wires. Full duplex – allows communication in two directions simultaneously.
Electrical connections	3 wires, Tx, Rx and common	3 wires, A, B and C (Common)	5 wires. A', B', A, B and C.
No. of drivers and receivers per line	1 driver 1 receiver	1 driver 31 receivers	1 driver 10 receivers
Maximum data rate	20k bits/s	10M bits/s	10M bits/s
Maximum cable length	50ft, (15M)	4000ft, (1200M)	4000ft, (1200M)

Notes:

1. EIA232C is abbreviated to EIA232 in this manual. The EIA232 standard allows a single instrument to be connected to a PC, a Programmable Logic Controller, or similar devices using a cable length of less than 15m.
2. The EIA485 standard allows one or more instruments to be connected (multi-dropped) using a three wire connection, with cable length of less than 1200M. 31 Instruments and one driver may be connected in this way. EIA485 is a balanced two-wire transmission system, which means that it is the difference in voltage between the two wires that carries the information rather than the voltage relative to ground or earth. One polarity of voltage indicates logic 1, the reverse polarity indicates logic 0. The difference must be at least +200mV but any voltage between +12 and -7 will allow correct operation. Balanced transmission is less prone to interference or pickup and should be used in preference to EIA232 in noisy environments. Although EIA485 is commonly referred to as a 'two wire' connection, a ground return/shield connection is provided as a 'common' connection for EPower controller units, and in general this should be used in installations to provide additional protection against noise.
3. The 3-wire EIA485 should be used where possible for new installations where multi-drop capability is required.
4. Some instruments use a terminal marking that is different from that used in the EIA485 standard. The table below compares this marking.

EIA485 3-Wire standard	Instrument		EIA485 5-Wire standard	Instrument	
			A'	RX+	RxA
			B'	RX-	RxB
A	A(+)	Rx	A	TX+	TxA
B	B(-)	Tx	B	TX-	TxB
C	Common (Com)	Common (Com)	C & C'	Common (Com)	Common (Com)

Notes:

1. EPower controller units operate in a half duplex mode that does not allow the simultaneous transmission and reception of data. Data is passed by an alternating exchange.
2. Most PCs provide an EIA232 port for digital communications. The KD485 Communications Interface unit is recommended for interfacing to EIA485. The limitation of 32 devices is overcome by splitting larger networks into segments that are electrically isolated. The KD485 may be used to buffer an EIA485 network when it is required to communicate with more than 32 instruments on the same bus. It may also be used to bridge 3-wire EIA485 to 5-wire EIA485 network segments. Should it be necessary to integrate EPower controllers into an existing system using 5-wire EIA485 the Series 2000 Communications Handbook shows details of recommended wiring. See "Interconnection Diagram 3-Wire EIA485" on page 23 for wiring information for this unit, or refer to KD485 Installation and Operation Handbook.

Cable Selection

This section provides general information regarding the type of cable which should be used in a serial communications system.

The cable selected for the digital communications network should have the following electrical characteristics:

- Less than 100 ohm / km nominal dc resistance. Typically 24 AWG or thicker.
- Nominal characteristic impedance at 100 kHz of 100 ohms.
- Less than 60 pF / m mutual pair capacitance, (the capacitance between two wires in a pair).
- Less than 120 pF / m stray capacitance, (the capacitance between one wire and all others connected to ground).
- Use twisted pair cables with screen.

The selection of a cable is a trade off between cost and quality factors such as attenuation and the effectiveness of screening. For applications in an environment where high levels of electrical noise are likely, use a cable with a copper braid shield, (connect the shield to a noise free ground). For applications communicating over longer distances, choose a cable that also has low attenuation characteristics.

The following list is a selection of cables suitable for EIA485 communication systems, listed in order of decreasing quality.

Cables marked '**' are suitable for use with the wiring descriptions that follow.

Cables marked '***' use a different colour coding from that used in the wiring descriptions.

Belden No	Description
9842	2 twisted pairs with aluminium foil screen plus a 90% coverage copper screen **
9843	3 twisted pairs with aluminium foil screen plus a 90% coverage copper screen **
9829	2 twisted pairs with aluminium foil screen plus a 90% coverage copper screen

Belden No	Description
9830	3 twisted pairs with aluminium foil screen plus a 90% coverage copper screen *
8102	2 twisted pairs with aluminium foil screen plus a 65% coverage copper screen
8103	3 twisted pairs with aluminium foil screen plus a 65% coverage copper screen *
9729	2 twisted pairs with aluminium foil screen
9730	3 twisted pairs with aluminium foil screen *

Wiring General

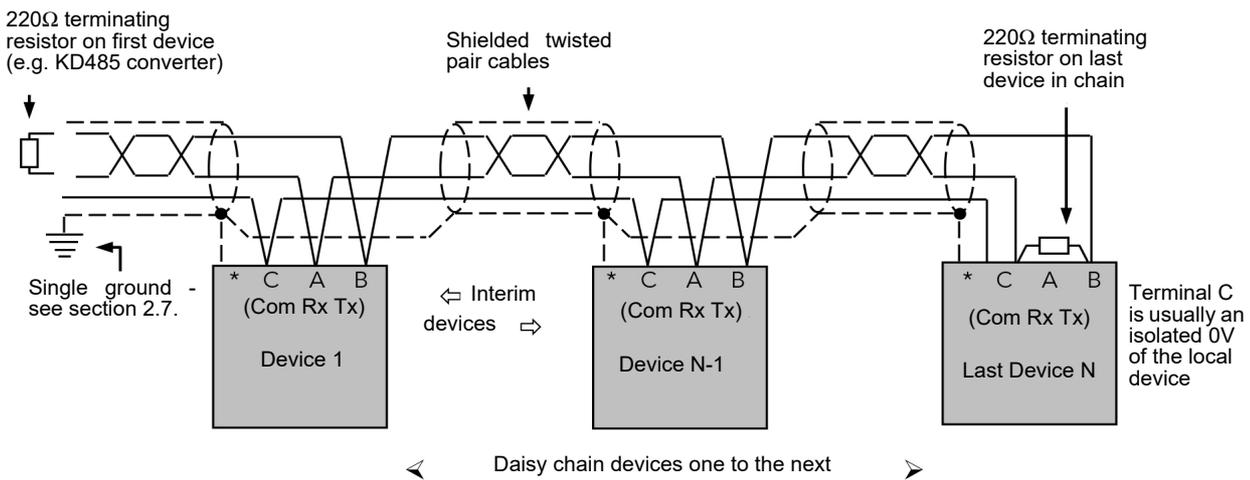
Route communications cables in separate trunking to power cables. Power cables are those connecting power to instruments, relay or triac, ac supplies and wiring associated with external switching devices such as contactors, relays or motor speed drives.

Communication cables may be routed with control signal cables if these signal cables are not exposed to an interference source. Control signals are the analogue or logic inputs and analogue or logic outputs of any control instrument.

Do not use redundant wires in the communications cable for other signals.

Ensure cable runs have sufficient slack to ensure that movement does not cause abrasion of the insulating sheath. Do not over tighten cable clamps to avoid accidental multiple grounding of the screen conductors.

Ensure that the cable is 'daisy chained' between instruments, i.e. the cable runs from one instrument to the next to the final instrument in the chain.



* On some devices a terminal may be supplied for connection of the shield. Omit this link if the terminal is not available.

Figure 5 Generic Principle for Interconnecting 3-wire (+ screen) EIA485 Compliant Communication Hardware

Precautions

In some installations, where excessive static build up may be experienced, it is advisable to add a high value resistor (1M Ω , for example) between the common connection and ground. For EPower units a bleed resistor of 2M Ω is already built into every communications port so that this external resistor is not necessary.

Note: EPower controller units are interconnected using RJ45 connectors. The interconnection cable must be fitted with plugs which have an outer metallic enclosure coupled to the screen of the cable (i.e. the metallic body is crimped to the wire screen). Suitable cables are available from Eurotherm as detailed in "Wiring 3-Wire EIA485" on page 22.

Grounding

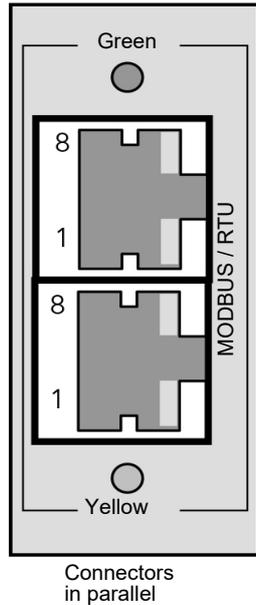
The EIA standard suggests that both ends of the cable shield be connected to safety ground (earth). If such a course is followed, care should be taken to ensure that differences in local earth potentials do not allow circulating currents to flow. These can not only induce large common mode signals in the data lines, resulting in communications failure, but can also produce overheating in the cable. Where doubt exists, it is recommended that the shield be grounded (earthed) at only one point. The wire length (shield to ground) should be kept as short as possible. Further information may be found in the EMC Installation Guide part number HA025464 which may be downloaded from www.eurotherm.com.

Digital Communications Hardware

Serial communications is provided by different modules which may be fitted into the comms slot. Pinouts for the various protocols are shown below:

Communications Pinouts

Modbus RTU



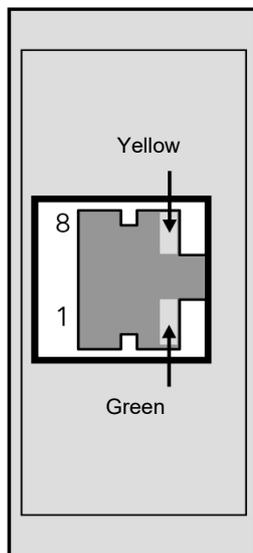
Pin	Signal (EIA485)
8	Reserved
7	Reserved
6	N/C
5	N/C
4	N/C
3	Isolated 0V
2	A
1	B

Internal connections:
 Pin 1 to 5V via 100k?
 Pin 2 to 0V via 100k?

LEDs:
 Green = Tx activity
 Yellow = Rx activity

Figure 6 Modbus RTU pinout

Modbus TCP (Ethernet 10baseT), Single Port



Pin	Function
8	N/C
7	N/C
6	Rx-
5	N/C
4	N/C
3	Rx+
2	Tx-
1	Tx+

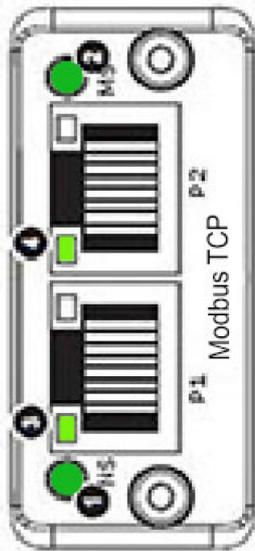
LEDs:
 Green = Tx activity
 Yellow = Network activity

Figure 7 Modbus TCP (Ethernet 10baseT) pinout, single port version

Modbus TCP (Dual Port)

Network status LED Indication (1)	
LED State	Interpretation
Off	No power or no IP address
Steady green	Module is in Process Active or Idle state
Flashing green	On-line, waiting for connection
Steady red	Duplicate IP address, or FATAL event
Flashing red	Process Active Timeout

Module status LED Indication (2)	
LED State	Interpretation
Off	No power
Steady green	Normal operation
Steady red	Major Fault (Exception state, fatal error etc.)
Flashing red	Minor fault in diagnostic object, IP conflict



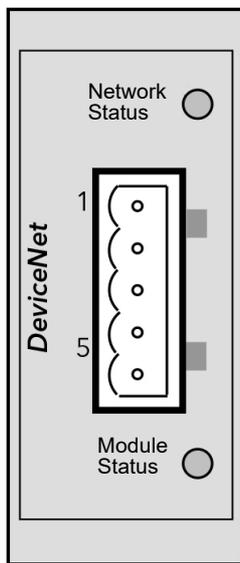
LINK LED (3, 4)	
LED State	Interpretation
Off	No Link, no activity
Steady Green	Link established
Flickering Green	Activity

Figure 8 Modbus TCP connector, dual port version

DeviceNet

Network status LED Indication	
LED State	Interpretation
Off	Off-line or no power
Steady green	On-line to 1 or more units
Flashing green	On-line no connections
Steady red	Critical link failure
Flashing red	1 or more connections timed out

Module status LED Indication	
LED State	Interpretation
Off	No power
Steady green	Operating normally
Flashing green	Missing or incomplete configuration
Steady red	Unrecoverable fault(s)
Flashing red	Recoverable fault(s)



Pin	Function
1	V- (negative bus supply voltage)
2	CAN_L
3	Cable shield
4	CAN_H
5	V+ (positive bus supply voltage)

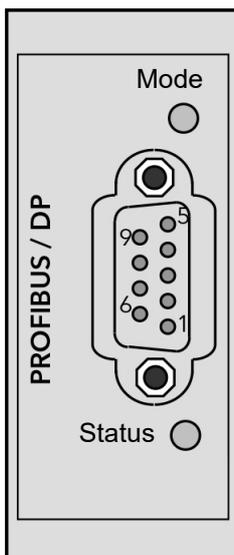
- Notes:**
1. See DeviceNet specification for power supply specification
 2. During startup, an LED test is performed, satisfying the DeviceNet standard

Figure 9 DeviceNet connector pinout

Profibus

Operation Mode Led Indication	
LED State	Interpretation
Off	Off-line or no power
Steady green	On-line, data exchange
Flashing green	On-line, clear
Red single flash	Parameterisation error
Red double flash	PROFIBUS configuration error

Status LED Indication	
LED State	Interpretation
Off	No power or not initialised
Steady green	Initialised
Flashing green	Diagnostic event present
Steady red	Exception error



Pin	Function	Pin	Function
9	N/C	5	Isolated ground
8	A (RxD-/TxD-)	4	RTS
7	N/C	3	B (RxD+/TxD+)
6	+5V (See note 1)	2	N/C
		1	N/C

Notes:

1. Isolated 5 Volts for termination purposes. Any current drawn from this terminal affects the total power consumption
2. The cable screen should be terminated to the connector housing

Figure 10 Profibus connector pinout

EtherNet/IP (Single Port)

LED Status Indication		A test sequence is performed on these LEDs during startup.	
MS	Module Status LED	NS	Network Status LED
Off	No power	Off	No power or no IP address
Green	Controlled by a scanner in the Run state	Green	On-line, one or more connections established (CIP Class 1 or 3)
Green flashing	Not configured or scanner in Idle state	Green flashing	On-line, no connections established
Red	Major fault (EXEPTION-state, FATAL error, etc)	Red	Duplicate IP address, FATAL error
Red flashing	Recoverable fault(s)	Red flashing	One or more connections timed out (CIP Class 1 or 3)

LED Status Indication	
LINK	Link/Activity LED
Off	No link, no activity
Green	Link established
Green flickering	Activity

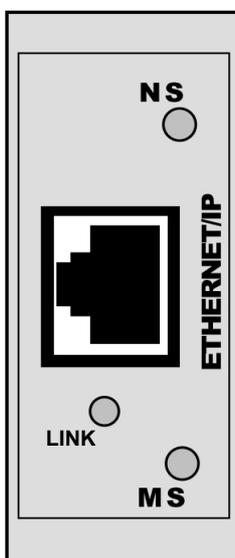
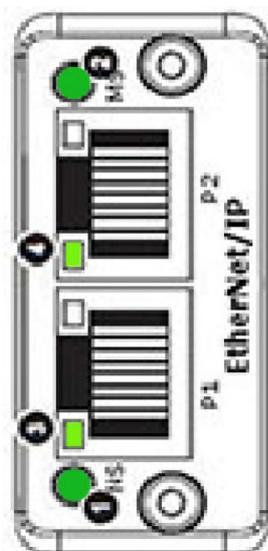


Figure 11 EtherNet/IP connector pinout, single port version

EtherNet/IP (Dual Port)

NS (Network status) LED (1)	
LED state	Interpretation
Off	No power or no IP address
Steady green	On-line; one or more connections established (CIP class 1 or 3)
Flashing green	On-line, no connections enabled
Steady red	Duplicate IP address ('fatal' error)
Flashing red	One or more connections timed out (CIP class 1 or 3)

MS (Module status) LED (2)	
LED state	Interpretation
Off	No power
Steady green	Controlled by a scanner in Run state
Flashing green	Not configured or scanner in Idle state
Steady red	Major fault (Exception state, fatal error etc.)
Flashing red	Recoverable fault



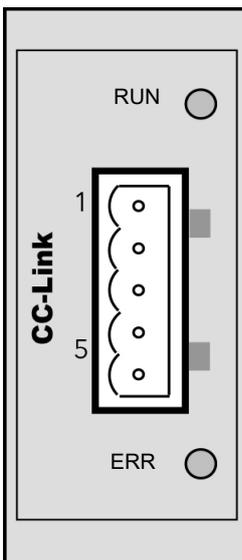
LINK LED (3, 4)	
LED state	Interpretation
Off	No link; no activity
Steady green	Link established
Flickering green	Activity in progress

Figure 12 EtherNet/IP connector, dual port version

CC-Link

LED Status Indication	
RUN LED	Interpretation
Off	No network participation, timeout status (no power)
Green	Participating, normal operation
Red	Major fault (FATAL error)

LED Status Indication	
ERROR LED	Interpretation
Off	No error detected (no power)
Red	Major fault (Exception or FATAL event)
Red, flickering	CRC error (temporary flickering)
Red flashing	Station number or Baud rate has changed since startup



Pin	Signal	Comment
1	DA	Positive EIA485 RxD/TxD
2	DB	Negative EIA485 RxD/TxD
3	DG	Signal ground
4	SLD	Cable shield
5	FG	Protective earth

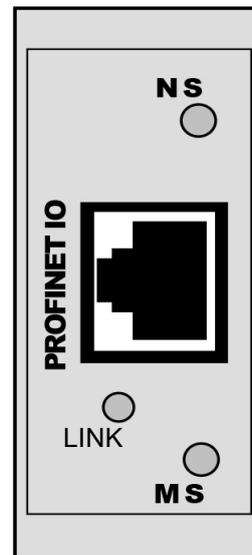
Figure 13 CC-Link connector pinout

PROFINET IO (Single Port)

LED Status Indication

NS (Network status) LED. A test sequence is performed on this LED during startup.		
LED state	Description	Interpretation
Off	Offline	- No power - No connection with I/O Controller
Steady Green	Online (RUN)	- Connection with I/O Controller established. - I/O Controller in 'Run' state.
Flashing Green	Online (STOP)	- Connection with I/O Controller established. - I/O Controller in 'Stop' state.

LINK LED		
LED state	Description	Interpretation
Off	No link	- No link - No communications present
Steady Green	Link	- Ethernet link established - No communications present
Flashing Green	Activity	- Ethernet link established - Communications present



MS (Module status) LED. A test sequence is performed on this LED during startup.		
LED state	Description	Interpretation
Off	Not initialised	- No power or the module is in 'SETUP' or 'NW_INIT' state
Green	Normal operation	- The module has shifted from the 'NW_INIT' state
Green, one flash	Diagnostic event(s)	- One or more Diagnostic Event present
Green, two flashes	Blink	- Used by engineering tools to identify the node on the network
Red	Exception error	- The module is in the 'EXCEPTION' state
Red, one flash	Configuration error	- The Expected Identification differs from the Real Identification
Red, two flashes	IP address error	- The IP address is not set
Red, three flashes	Station name error	- The Station Name is not set
Red, four flashes	Internal error	- The module has encountered a major internal fault

Figure 14 PROFINET connector pinout, single port version

PROFINET IO (Dual Port)

NS (Network status) LED (1)	
LED state	Interpretation
Off	No power or IP address
Steady green	On-line (RUN); connection with I/O controller established. Controller in 'Run' state
Flashing green	On-line (STOP); connection with IO controller established. Controller in 'Stop' state.

MS (Module status) LED (2)		
LED state	Interpretation	
Off	Not initialised	No power, or the module is in 'SETUP' or 'NW_INIT' state.
Green steady	Normal operation	The module has shifted from the 'NW_INIT' state.
Green 1 flash	Diagnostic event	One or more Diagnostic Event present.
Green 2 flash	Blink	Used by engineering tools to identify the node on the network.
Red steady	Exception error	The module is in the 'EXCEPTION' state.
Red1 flash	Configuration error	The Expected Identification differs from the Real Identification.
Red 2 flash	IP address error	The IP address is not set.
Red 3 flash	Station Name error	The Station name is not set.
Red 4 flash	Internal error	The module has detected a major internal fault.

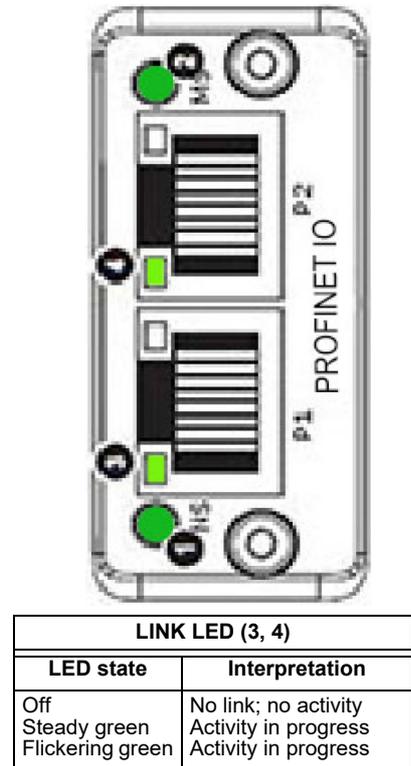


Figure 15 PROFINET connector, double port version

Wiring 3-Wire EIA485

The EIA485 standard allows one or more instruments to be connected (multi dropped) using a three wire connection, with cable length of less than 1200m. 31 instruments and one master may be connected. The communication line must be daisy chained from device to device and two RJ45 sockets are provided on each EPower controller unit to simplify this (see "Modbus RTU" on page 17).

1. Suitable cables are available from Eurotherm and can be ordered as:

2500A/CABLE/MODBUS/RJ45/RJ45/0M5 Cable 0.5 metres long

2500A/CABLE/MODBUS/RJ45/RJ45/3M0 Cable 3.0 metres long

The plugs fitted to these cables are provided with an outer metallic shell which is connected to the wire screen of the cable. This type of cable must be used for EMC compliance.

2. All network communications lines must be terminated using the appropriate impedance as shown in [Figure 5](#). To simplify installation a plug-in line terminator is available, order code 2500A/TERM/MODBUS/RJ45 which can be plugged into the free socket in the last EPower controller unit in the chain. This provides correct terminating resistor values for CAT-5 cable.

The terminator is shown below:

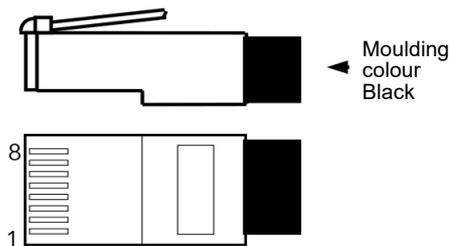


Figure 16 Line termination for Modbus

The moulding for the Modbus terminator is coloured BLACK.

3. To use EIA485, buffer the EIA232 port of the PC with a suitable EIA232/EIA485 converter. The Eurotherm Controls KD485 Communications Adapter unit is recommended for this purpose. The use of a EIA485 board built into the computer is not recommended since this board may not be isolated, which may cause noise problems or damage to the computer, and the RX terminals may not be biased correctly for this application.

Either cut a patch cable and connect the open end to the KD485 converter or, using twin screened cable as detailed in "Cable Selection" on page 14, crimp an RJ45 plug on the EPower controller end.

The PC serial port may be either 9 pin (most common) or 25 pin. The table below lists the correct connections for either of these ports.

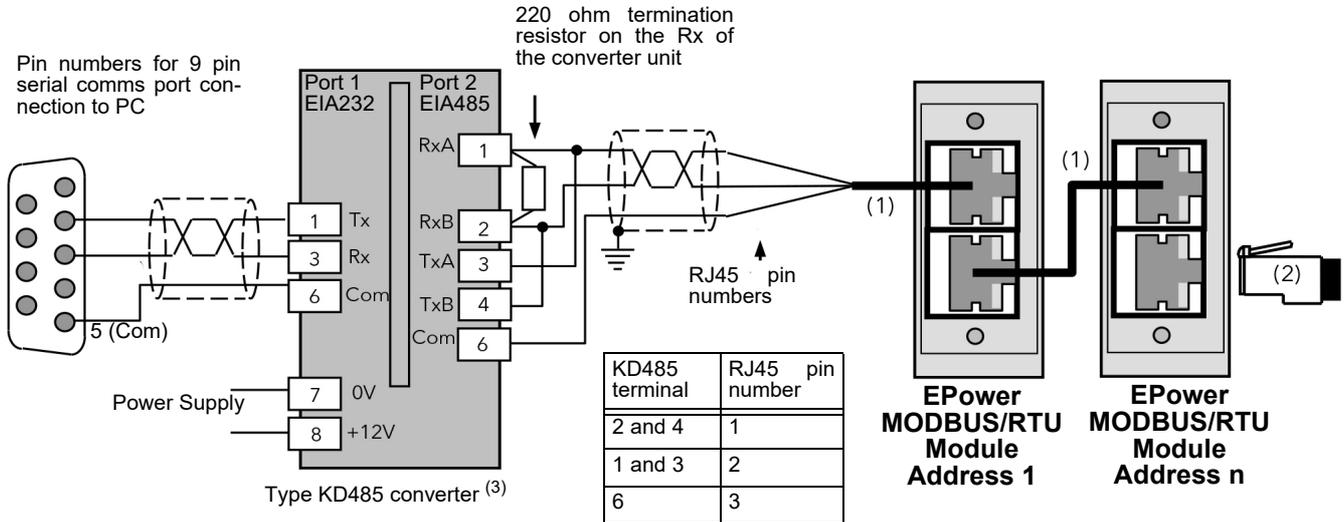
Standard Cable Colour	PC socket pin no		PC Function	Instrument Terminal	Instrument Function
	* 9 pin	25 pin			
White	2	3	Receive (RX)	HF	Transmit (TX)
Black	3	2	Transmit (TX)	HE	Receive (RX)
Red	5	7	Common	HD	Common
Link together	1 4 6	6 8 11	Rec'd line sig. detect Data terminal ready Data set ready		
Link together	7 8	4 5	Request to send Clear to send		
Screen		1	Ground		

Interconnection Diagram 3-Wire EIA485

Example 1: Two EPower controller units are connected to a PC via a KD485 communications converter.

The interconnecting cables between EPower controllers use RJ45 to RJ45 cables (1) as described in "Wiring 3-Wire EIA485" on page 22. They should be fitted with plugs provided with an outer metallic shell with the shell connected to the wire screen of the cable. This type of cable must be used to maintain EMC compliance.

To connect an EPower controller to the communications converter an RJ45 to open end cable is required. It may be convenient to cut a standard RJ45/RJ45 cable in order to connect it to the KD485 terminals. In this case make sure that the shield is connected to ground (chassis) with a connection as short as possible to the convertor (see also "Grounding" on page 16).



1. RJ45 Patch cables
2. Modbus Terminator (see previous section)
3. The KD485 Installation and Operation Handbook supplied with the unit provides further connection and configuration details.

Figure 17 EIA485 Three Wire Interconnections (EPower controller units only)

Example 2: It is also likely that other units such as 3500 controllers may also be required on the network. The diagram below shows an example of a 3000 series instrument connected to the above network. Instruments may be connected at any point in the network but daisy chaining, grounding of the shield and correct line terminations (Figure 18) should be observed.

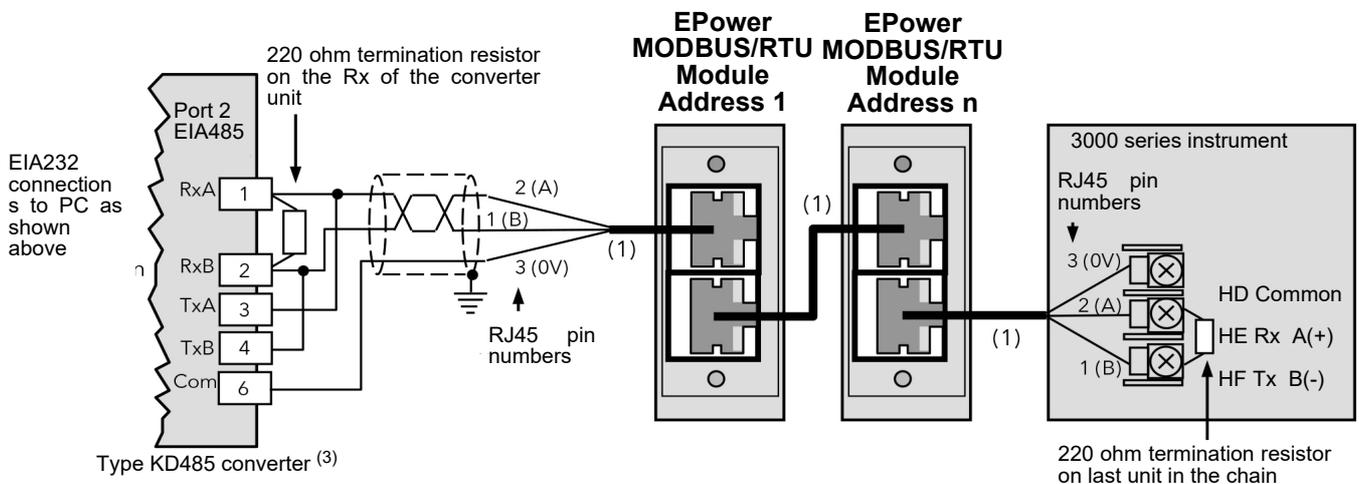


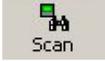
Figure 18 EIA485 3-wire (EPower controller and Series 3000 units)

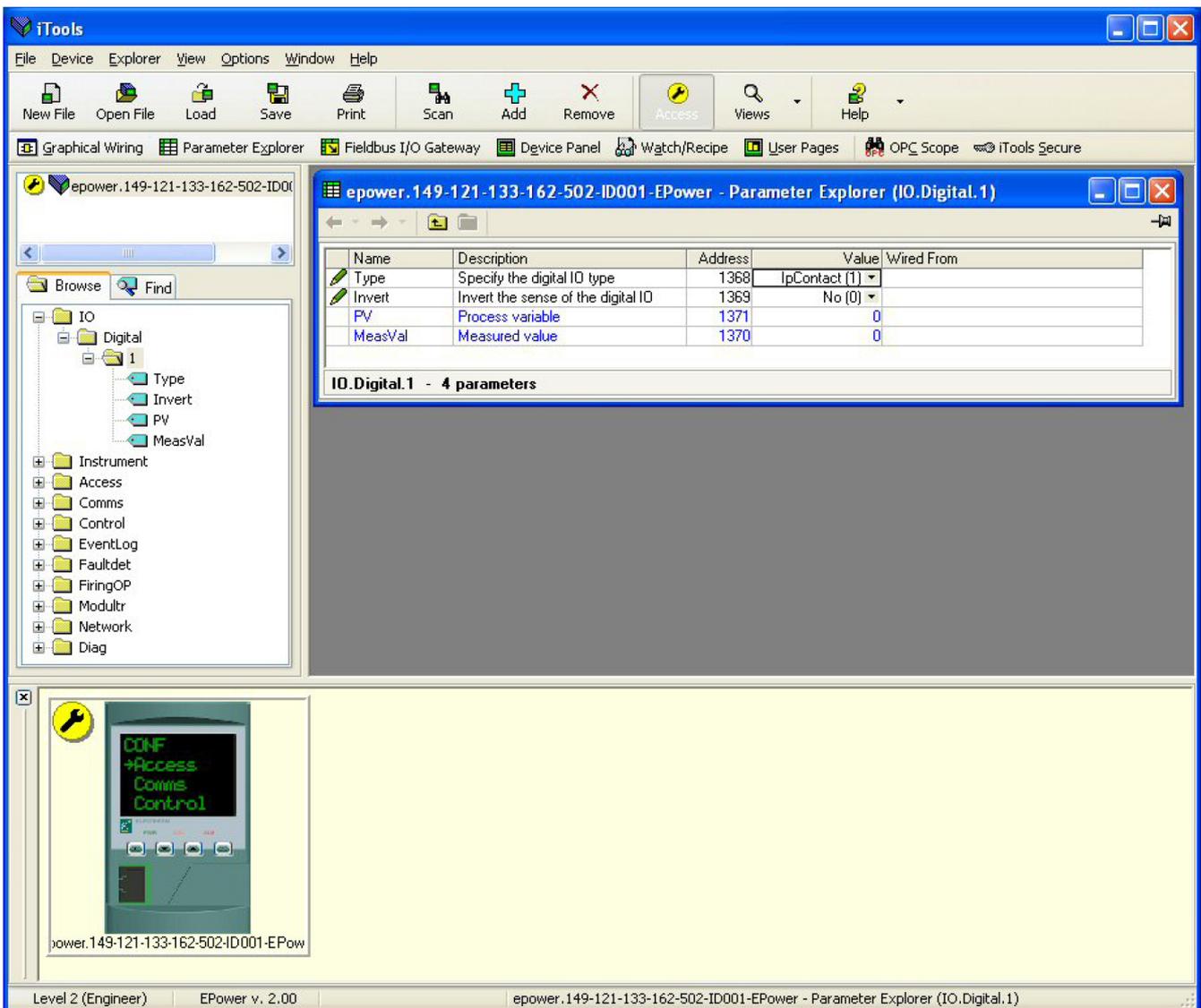
It is also possible that EPower controller units are to be used in existing installations which already use other transmission standards such as EIA422 (EIA485 5-wire). If this is the case refer to the 2000 Series Communications Handbook, HA026230 for further wiring details.

iTools

iTools is a software configuration package which is supplied on a CD with EPower controller units. Details of iTools can be found in the iTools Help Manual Part No. HA028838 which is available on the CD or may be downloaded from www.eurotherm.com.

With iTools loaded on the PC and the address, baud rate, parity and delay set

appropriately, press  EPower controller units connected to the PC will be detected and displayed generally as shown below.



The version of iTools supplied with the EPower controller unit will correspond with the firmware version of the EPower controller unit. In the event that a later firmware version of EPower controller is used with an older version of iTools the instrument view may display three question marks (???). This shows that the unit has been detected but that the versions do not match. This may generally be solved by downloading the latest copy of iTools from www.eurotherm.com.

Modbus Parameters

The following parameters are available. They may be viewed/set in the Engineer Level or Configuration Level. The procedure is described in the EPower controller User Guide HA179769.

Comms Identity

Read only in all levels. Displays the type of communications board fitted, i.e. RS-485 (EIA485).

Protocol

Read only in all levels. Displays the transmission protocol, i.e. Modbus

Unit Address

Each EPower controller slave must have its own unique address. Read/Write in Engineer and Configuration levels, the settable range is 1 to 254.

Baud Rate

Read/Write in Configuration level. The baud rate of a communications network specifies the speed at which data is transferred between the instrument and the master. Generally the baud rate should be set as high as possible to allow maximum speed of operation. This will depend to some extent on the installation and the amount of electrical noise the communications link is subject to. EPower controller units are capable of reliably operating at 19,200 baud under normal circumstances and assuming correct line termination, see "Wiring 3-Wire EIA485" on page 22.

Although the baud rate is an important factor, when calculating the speed of communications in a system it is often the 'latency' ("Latency" on page 40) between a message being sent and a reply being started that dominates the speed of the network. This is the amount of time the instrument requires on receiving a request before being able to reply.

For example, if a message consists of 10 characters (transmitted in 10msec at 9600 Baud) and the reply consists of 10 characters, then the transmission time would be 20 msec. However, if the latency is 20msec, then the transmission time has become 40msec. Latency is typically higher for commands that write to a parameter than those that read, and will vary to some degree depending on what operation is being performed by the instrument at the time the request is received and the number of variables included in a block read or write. As a rule, latency for single value operations will be between 5 and 20 msec, meaning a turnaround time of about 25-40msec.

If throughput is a problem, consider replacing single parameter transactions with Modbus block transactions, and increase the baud rate to the maximum reliable value in your installation.

All units in a system must be set to the same baud rate. Most SCADA packages will automatically adjust to the baud rate set in the slaves but if this is not the case set the baud rate to match that set in the package.

The settable range is 4800, 9600, 19,200.

Parity

Read/Write in Configuration level. Parity is a method of ensuring that the data transferred between devices has not been corrupted. It is the lowest form of integrity in the message and ensures that a single byte contains either an even or an odd number of ones or zeros in the data.

In industrial protocols, there are usually layers of checking to ensure that the first byte transmitted is good and then that the message transmitted is good. Modbus applies a CRC (Cyclic Redundancy Check, see [page 32](#)) to the data to ensure that the packet of data is not corrupted. Thus there is usually no benefit in using odd or even parity, and since this also increases the number of binary bits transmitted for any messages, it decreases throughput.

Value Options:-

NONE (0): No Parity Enabled. This is the normal default setting for Eurotherm instruments.

EVEN (1): Even Parity

ODD (2): Odd Parity

(0), (1) or (2) indicates the enumeration of the value.

Delay

Read/Write in Configuration level. Tx Delay Time. In some systems it is necessary to introduce a delay between the instrument receiving a message and making its reply to ensure a guaranteed gap

For example, this is sometimes needed by communications converter boxes which require a period of silence on the transmission to switch the direction of their drivers.

Value Options:- OFF (0): No delay; ON (1): Set this variable to 'On' to insert a guaranteed 10msec delay between the transmission being received and the response being made.

Modbus and Jbus Protocol

This chapter introduces the principles of the Modbus and JBus communication protocols. Note that in EPower controller units, these two protocols are identical but the J-bus address is displaced by '1' relative to Modbus. Both will be referred to as Modbus for the descriptions that follow.

Protocol Basics

A data communication protocol defines the rules and structure of messages used by all devices on a network for data exchange. This protocol also defines the orderly exchange of messages, and the detection of errors.

Modbus defines a digital communication network to have only one MASTER and one or more SLAVE devices. Either a single or multi-drop network is possible. The two types of communications networks are illustrated in the diagram below:

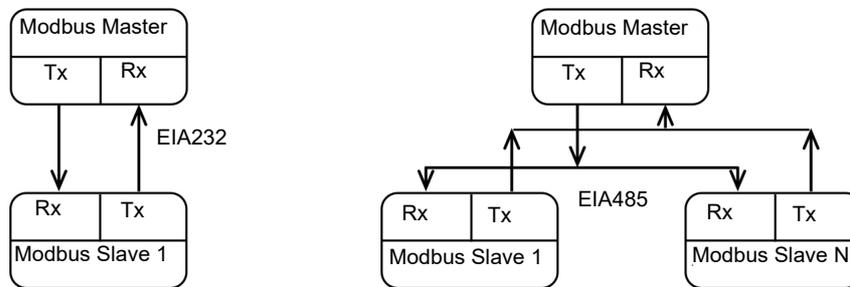


Figure 19 Single Serial Link

Multi Drop Serial Link

A typical transaction will consist of a request sent from the master followed by a response from the slave.

The message in either direction will consist of the following information:

Device Address	Function Code	Data	Error Check Data	End of Transmission
----------------	---------------	------	------------------	---------------------

- Each slave has a unique 'device address'.
- The device address 0 is a special case and is used for messages broadcast to all slaves. This is restricted to parameter write operations.
- EPower controller supports a subset of Modbus function codes.
- The data will include instrument parameters referenced by a 'parameter address'.
- Sending a communication with a unique device address will cause only the device with that address to respond. That device will check for errors, perform the requested task and then reply with its own address, data and a check sum.
- Sending a communication with the device address '0' is a broadcast communication that will send information to all devices on the network. Each will perform the required action but will not transmit a reply.

Typical Transmission Line Activity

This diagram is to illustrate typical sequence of events on a Modbus transmission line.

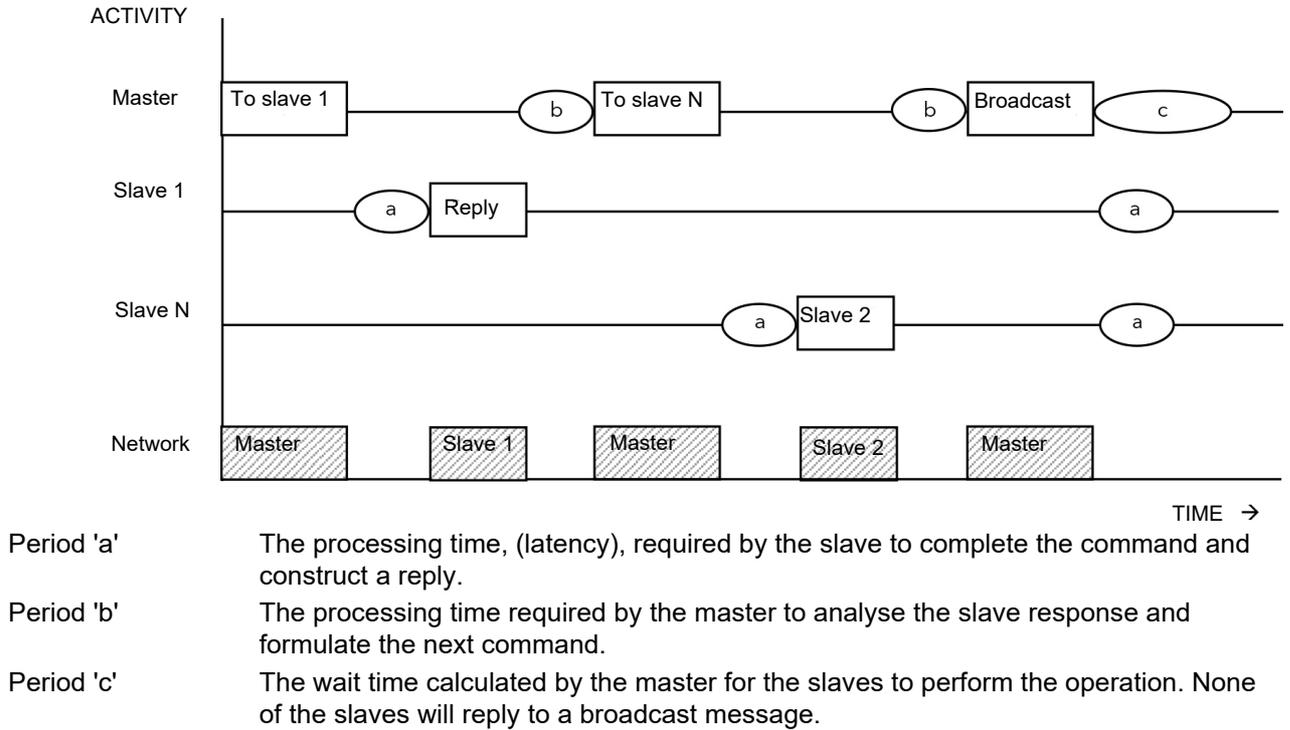


Figure 20 Modbus Timing Diagram

For a definition of the time periods required by the network, refer to 'Wait Period' in section 4.18 'Error Response'.

Device Address

Each slave has a unique 8 bit device address. The Modbus Protocol defines the address range limits as 1 to 247. EPower controller units support an address range of 1 to 254.

Parameter Address

Data bits or data words exchange information between master and slave devices. This data consists of parameters. All parameters communicated between master and slaves have a 16 bit parameter address.

The Modbus parameter address range is 0001 to FFFF.

Parameter addresses for EPower controller units is given in the User Guide HA17969.

Parameter Resolution

Modbus (and JBus) protocol limits data to 16 bits per parameter. This reduces the active range of parameters to 65536 counts. In EPower controller units this is implemented as -32767 (8001h) to +32767 (7FFFh).

The protocol is also limited to integer communication only. EPower controller units allow full resolution. In full resolution mode the decimal point position will be implied so that 100.01 would be transmitted as 10001. From this, and the 16 bit resolution limitation, the maximum value communicable with 2 decimal place resolution is 327.67. The parameter resolution will be taken from the slave user interface, and the conversion factor must be known to both master and slave when the network is initiated.

EPower controller units provide a special sub-protocol for accessing full resolution floating point data. This is described in "Modbus Advanced Topics" of this manual.

Reading of Large Numbers

Large numbers being read over digital communications are scaled. For example, Setpoint can have the maximum value of 99,999 and is read as nnn.nK or 100,000 = 100.0K and 1,000,000 = 1000.0K.

Mode of Transmission

The mode of transmission describes the structure of information within a message and the number coding system used to exchange a single character of data.

The Modbus (and JBus) Protocols define a mode of transmission for both ASCII and RTU modes of transmission. EPower controller units only support the RTU mode of transmission.

The RTU definition of the mode of transmission for a single character is;

A start bit, eight data bits, a parity bit and one or two stop bits

All EPower controller units use 1 stop bit.

Parity may be configured to be NONE, ODD or EVEN.

If parity is configured to be NONE, no parity bit is transmitted.

The RTU mode of transmission for a single character is represented as follows:



Message Frame Format

A message consists of a number of characters sequenced so that the receiving device can understand. This structure is known as the message frame format.

The following diagram shows the sequence defining the message frame format used by Modbus and Jbus:

Frame Start	Device Address	Function Code	Data	CRC	EOT
3 bytes	1 byte	1 byte	n bytes	2 byte	3 bytes

The Frame Start is a period of inactivity at least 3.5 times the single character transmission time.

For example, at 9600 baud a character with 1 start, 1 stop and 8 data bits will require a 3.5ms frame start.

This period is the implied EOT of a previous transmission.

The Device Address is a single byte (8-bits) unique to each device on the network.

Function Codes are a single byte instruction to the slave describing the action to perform.

The Data segment of a message will depend on the function code and the number of bytes will vary accordingly.

Typically the data segment will contain a parameter address and the number of parameters to read or write.

The Cyclic Redundancy Check, (CRC) is an error check code and is two bytes, (16 bits) long.

The End of Transmission segment, (EOT) is a period of inactivity 3.5 times the single character transmission time. The EOT segment at the end of a message indicates to the listening device that the next transmission will be a new message and therefore a device address character.

Cyclic Redundancy Check

The Cyclic Redundancy Check, (CRC) is an error check code and is two bytes, (16 bits) long. After constructing a message, (data only, no start, stop or parity bits), the transmitting device calculates a CRC code and appends this to the end of the message. A receiving device will calculate a CRC code from the message it has received. If this CRC code is not the same as the transmitted CRC there has been a communication error. EPower controller units do not reply if they detect a CRC error in messages sent to them.

The CRC code is formed by the following steps:

1. Load a 16 bit CRC register with FFFFh.
2. Exclusive OR (\oplus) the first 8 bit byte of the message with the with the high order byte of the CRC register.
Return the result to the CRC register.
3. Shift the CRC register one bit to the right.
4. If the over flow bit, (or flag), is 1, exclusive OR the CRC register with A001 hex and return the result to the CRC register.

Note: If the overflow flag is 0, repeat step 3.

5. Repeat steps 3 and 4 until there have been 8 shifts.
6. Exclusive OR the next 8 bit byte of the message with the high order byte of the CRC register.
7. Repeat step 3 through to 6 until all bytes of the message have been exclusive OR with the CRC register and shifted 8 times.
8. The contents of the CRC register are the 2 byte CRC error code and are added to the message with the most significant bits first.

The flow chart below illustrates this CRC error check algorithm.

The ' \oplus ' symbol indicates an 'exclusive OR' operation. 'n' is the number of data bits.

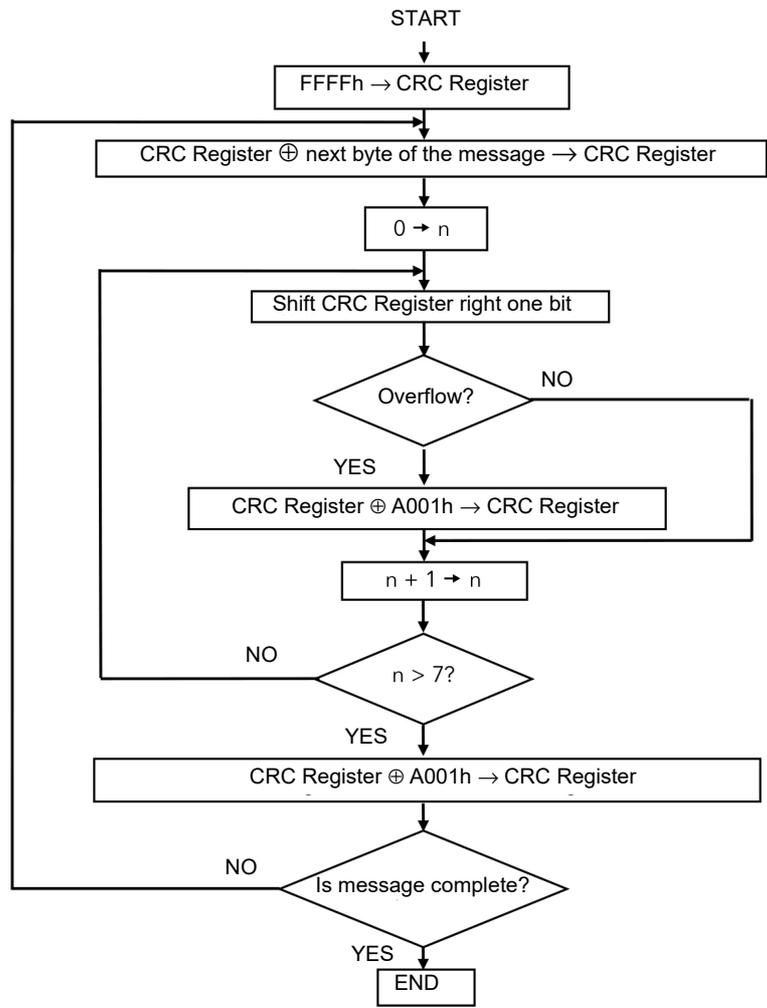


Figure 21 CRC Flow Chart

Example of a CRC Calculation

This example is a request to read from the slave unit at address 02, the fast read of the status (07).

Function	16 Bit Register				Carry flag
	LSB		MSB		
Load register with FFFF hex	1111	1111	1111	1111	0
First byte of the message (02)			0000	0010	
Exclusive OR	1111	1111	1111	1101	
1st shift right	0111	1111	1111	1110	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1101	1111	1111	1111	
2nd shift right	0110	1111	1111	1111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1100	1111	1111	1110	
3rd shift right	0110	0111	1111	1111	0
4th shift right (carry = 0)	0011	0011	1111	1111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1001	0011	1111	1110	

Function	16 Bit Register				Carry flag
	LSB		MSB		
5th shift right	0100	1001	1111	1111	0
6th shift right (carry = 0)	0010	0100	1111	1111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1000	0100	1111	1110	
7th shift right	0100	0010	0111	1111	0
8th shift right (carry = 0)	0010	0001	0011	1111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1000	0001	0011	1110	
Next byte of the message (07)			0000	0111	
Exclusive OR (shift = 8)	1000	0001	0011	1001	
1st shift right	0100	0000	1001	1100	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1110	0000	1001	1101	
2nd shift right	0111	0000	0100	1110	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1101	0000	0100	1111	
3rd shift right	0110	1000	0010	0111	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1100	1000	0010	0110	

Function	16 Bit Register				Carry flag
	LSB		MSB		
4th shift right	0110	0100	0001	0011	0
5th shift right (carry = 0)	0011	0010	0000	1001	1
A001	1010	0000	0000	0001	
Exclusive OR (carry = 1)	1001	0010	0000	1000	
6th shift right	0100	1001	0000	0100	0
7th shift right (carry = 0)	0010	0100	1000	0010	0
8th shift right (carry = 0)	0001	0010	0100	0001	0
CRC error check code	12h		41h		

The final message transmitted, including the CRC code, is as follows;

Device address		Function code		CRC MSB		CRC LSB	
02h		07h		41h		12h	
0000	0010	0000	0111	0100	0001	0001	0010
↑ First bit		Transmission order				Last bit ↑	

Example of a CRC Calculation in the 'C' Language

This routine assumes that the data types 'uint16' and 'uint8' exists. These are unsigned 16 bit integer (usually an 'unsigned short int' for most compiler types) and unsigned 8 bit integer (unsigned char). 'z_p' is a pointer to a Modbus message, and z_message_length is its length, excluding the CRC. Note that the Modbus message will probably contain 'NULL' characters and so normal C string handling techniques will not work.

```
uint16 calculate_crc(byte *z_p, uint16 z_message_length)
/* CRC runs cyclic Redundancy Check Algorithm on input z_p */
/* Returns value of 16 bit CRC after completion and */
/* always adds 2 crc bytes to message */
/* returns 0 if incoming message has correct CRC */

{
  uint16 CRC= 0xffff;
  uint16 next;
  uint16 carry;
  uint16 n;
  uint8 crch, crcl;

  while (z_message_length--) {
    next = (uint16)*z_p;
    CRC ^= next;
    for (n = 0; n < 8; n++) {
      carry = CRC & 1;
      CRC >>= 1;
      if (carry) {
        CRC ^= 0xA001;
      }
    }
    z_p++;
  }
  crch = CRC / 256;
  crcl = CRC % 256;
  z_p[z_message_length++] = crcl;
  z_p[z_message_length] = crch;
  return CRC;
}
```

Example of a CRC Calculation in Basic Language

```
Function CRC (message$) as long
'' CRC runs cyclic Redundancy Check Algorithm on input message$
'' Returns value of 16 bit CRC after completion and
'' always adds 2 crc bytes to message
'' returns 0 if incoming message has correct CRC

'' Must use double word for CRC and decimal constants

crc16& = 65535
FOR c% = 1 TO LEN(message$)
  crc16& = crc16& XOR ASC(MID$(message$, c%, 1))
  FOR bit% = 1 TO 8
    IF crc16& MOD 2 THEN
      crc16& = (crc16& \ 2) XOR 40961
    ELSE
      crc16& = crc16& \ 2
    END IF
  NEXT BIT%
NEXT c%
crch% = CRC16& \ 256: crcl% = CRC16& MOD 256
message$ = message$ + CHR$(crcl%) + CHR$(crch%)
CRC = CRC16&
END FUNCTION CRC
```

Function Codes

Function codes are a single byte instruction to the slave describing the action to perform.

The following function codes are supported by EPower controller units:

Function code	Function
03 or 04	Read n words
06	Write a word
08	Loopback
16	Write n words

It is recommended that function code 3 is used for reads and function code 16 is used for writes. This includes Boolean data. Other codes are supplied for purposes of compatibility.

Data words exchange information between master and slave devices. This data consists of parameters.

Parameter definitions for the EPower controller units are provided in the EPower controller User Guide.

The sections that follow explain the message frame format for each function code.

Read N Words

This gives the ability to read a sequential series of parameters in a single transaction. It is necessary to define both the address of the first parameter to read and the number of words to read following the first address.

Function code: 03 or 04, (03h or 04h)
 Command:

Device address	Function code 03 or 04	Address of first word		Number of words to read		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The maximum number of words that may be read is 125
 Reply:

Device address	Function code 03 or 04	Number of bytes read	Value of the first word		Value of the last word		CRC	
1 byte	1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Example:

From EPower controller slave at device address 2, read 2 words from parameter address 039B (h) 923 (dec). This parameter is Control 1 Main PV which is followed by 039C (h) 924 (dec), Control 1 Main SP.

Command:

Device address	Function code	Address of first word		Number of words to read		CRC	
02	03	03	9B	00	02	B5	93

Reply: (EPower controller units are configured with full resolution and PV = 18.3, SP = 21.6)

Device address	Function code 03 or 04	Number of bytes read	Value of the first word		Value of the last word		CRC	
02	03	04	00	B7	00	D8	79	4F

As the decimal point is not transmitted, the master must scale the response;
183 = 18.3, 216 = 21.6.

Write A Word

Function code: 06, (06h)

Command:

Device address	Function code 06	Address of word		Value of word		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Reply:

Device address	Function code 06	Address of word		Value of word		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The reply to function 06 is the same as the command. See "Error Response" on page 39 on 'Error Response' for details of the reply if the operation fails.

Example:

Write to an EPower controller slave at device address 2 and change the setpoint to 25.0 (address 039D). The instrument is configured with full resolution, therefore the required value is 250.

Command:

Device address	Function code	Address of word		Value of word		CRC	
02	06	03	9D	00	FA	98	10

Reply:

Device address	Function code	Address of word		Value of word		CRC	
02	06	03	9D	00	FA	98	10

Diagnostic Loopback

Function code: 08, (08h)

This function provides a means of testing the communications link by means of a 'loopback' operation. The data sent to the instrument is returned unchanged. Only diagnostic code 0 from the Gould Modicon Specification is supported.

Command:

Device address	Function Code 08	Diagnostic Code 0000		Loopback Data		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Reply:

The reply to function 08 is the same as the command

Example:

Perform a loopback from an EPower controller unit at address 2, using a data value of 1234 (h).

Command:

Device address	Function Code 08	Diagnostic Code 0000		Loopback Data		CRC	
02	08	00	00	12	34	ED	4F

Reply:

Device address	Function Code 08	Diagnostic Code 0000		Loopback Data		CRC	
02	08	00	00	12	34	ED	4F

Write N Words

This gives the ability to write a sequential series of parameters in a single transaction. It is necessary to define both the address of the first parameter to write and the number of words following the first address.

Function code: 16, (10h)

Command:

Device address	Function code 10	Address of first word		Number of words to write		Number of data bytes (n)	Data		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	1 byte	MSB	LSB	MSB	LSB

The maximum number of words that can be transmitted is 125 words, which corresponds to 250 bytes of data

The first two bytes are data with the required value of the first parameter, MSB first. Following pairs of bytes are data for the consecutive parameter addresses.

NB: Blocks of data written using Modbus function 16 containing values in positions corresponding to the addresses of unconfigured parameters are not generally rejected, although the values of any unconfigured parameters are discarded. This allows relatively large blocks of parameter data to be written in a single operation, even if the block contains a little 'empty' space. This is particularly useful for operations such as instrument cloning. However this also leads to a potential pitfall: if the block of data contains only a single parameter, and the destination address refers to an unconfigured or unused Modbus address, the write operation will appear to be successful, although the instrument will have discarded the value.

Attempts to write to read only parameters over Modbus, even when they are embedded within a block of data, will be rejected with a Modbus 'data error'. Any subsequent values in the block will also be discarded.

Reply:

Device address	Function code	Address of first word		Number of words written		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Example: Write to the EPower controller slave at device address 2 (configured with full resolution).

Control 1 Main PV = 12.3 (123) parameter address 039B (h)
 Control 1 Main Setpoint = 15.0 (150) parameter address 039C (h)
 Control 1 Main Transfer PV = 25.0 (250) parameter address 039D (h)

Command:

Device address	Function code	Address of first word		Number of words to write		Number of data bytes	Data	CRC	
02	10	03	9B	00	03	06	See below	1F	FA

Data (123) for address 039B		Data (150) for address 039C		Data (250) for address 039D	
00	7B	00	96	00	FA

Reply:

Device address	Function code	Address of first word		Number of words written		CRC	
02	10	03	9B	00	03	F1	90

Error Response

The Modbus and Jbus protocol define the response to a number of error conditions. A slave device is able to detect a corrupted command or, one that contains an incorrect instruction, and will respond with an error code.

With some errors the slave devices on the network are unable to make a response. After a wait period the master will interpret the failure to reply as a communication error. The master should then re-transmit the command.

Error Response Codes

A slave device that has detected a corrupted command or a command that contains an incorrect instruction, will respond with an error message. The error message has the following syntax.

Device address	Function code	Error response code	CRC	
1 byte	1 byte	1 byte	MSB	LSB

The Function code byte contains the transmitted function code but with the most significant bit set to 1.

[This is the result of adding 128 to the function code (Binary 10000000)].

The error response code indicates the type of error detected.

EPower controller units support the following error response codes:

Code	Error	Description
03	Illegal Data Value	The value referenced in the data field is not allowable in the addressed slave location

Wait Period

There are several errors for which the slave devices on the network are unable to make a response:

- If the master attempts to use an invalid address then no slave device will receive the message.
- For a message corrupted by interference, the transmitted CRC will not be the same as the internally calculated CRC. The slave device will reject the command and will not reply to the master.

After a wait period, the master will re-transmit the command.

The wait period should exceed the instrument latency plus the message transmission time. A typical wait period, for a single parameter read, is 100ms.

Latency

The time taken for an EPower controller unit to process a message and start the transmission of a reply is called the latency. This does not include the time taken to transmit the request or reply.

The parameter functions read 1 word (function 03h), write 1 word (function 06h), and loopback (function 08h) are processed within a latency of between 20 and 120ms (typically 90).

For the parameter functions read n words (function 03h) and write n words (function 16h) the latency is indeterminate. The latency will depend on the instrument activity and the number of parameters being transferred and will take from 20 to 500ms.

It is possible to artificially increase the latency by setting the 'User Comms Delay' parameter in the 'Comms' configuration list. This is sometimes required to allow a guaranteed gap between requests and responses, needed by some RS485 adaptors to switch from transmit to receive states.

Message Transmission Time

The time required to transmit a message will depend on the length of the message and the baud rate.

Message transmission time =

$$\frac{(\text{Number of bytes in the message} + 3.5) * \text{Number of bits per character}}{\text{Baud rate}}$$

To find the number of bytes, refer to the relevant function code. The three extra bytes are for the End of Transmission, (EOT), characters.

The number of bits per character will be ten, or eleven if a parity bit is used. (1 start bit, 8 data bits, an optional parity bit and 1 stop bit. See Mode of Transmission, "Mode of Transmission" on page 31).

For example reading a single word with the function code 03 at 19200 baud, (no parity bit):

$$\text{Command transmission time} = \frac{(8 + 3.5) * 10}{19200} = 6 \text{ ms}$$

$$\text{Reply transmission time} = \frac{(9 + 3.5) * 10}{19200} = 6.5 \text{ ms}$$

The wait period for this transaction will exceed 62.5 ms, (6 + 6.5 + 50.0).

Status Words

Status words group together commonly accessed parameters in convenient categories so that they may be read (or occasionally written to) as a single transaction. Their main use is to allow the most commonly required process conditions to be read quickly. Status Words in EPower controller are:

Fieldbus Comms Status

Bit No.	Description
0	Missing Mains
1	Thyristor Short Circuit
2	Open Thyristor
3	Fuse Blown
4	Over Temp
5	Network Dips
6	Frequency Fault
7	PB24V Fault
8	TLF
9	Chop Off
10	PLF
11	PLU
12	Volt Fault
13	Pre Temp
14	Over Current
15	Pwr Mod Watchdog Fault
16	Pwr Mod Comms Error
17	Pwr Mod Comms Timeout
18	Closed Loop
19	Transfer Active
20	Limit Active
21	Load Management Pr Over Ps
22	Output Fault
23-31	Reserved

Strategy Status Word

This is a bitmap parameter that indicates the status of the strategy. It is generally intended for internal use but can be accessed by the user. The description of each bit and its meaning when it is set is as follows:

Bit No.	Description
0	Network 1 is not firing
1	Network 1 is not synchronised
2	Network 2 is not firing
3	Network 2 is not synchronised
4	Network 3 is not firing
5	Network 3 is not synchronised
6	Network 4 is not firing
7	Network 4 is not synchronised
8	Strategy is in Standby Mode
9	Strategy is in Telemetry Mode
10-15	Reserved

Error Status Words

The Event Log is a 40 event FIFO table of instrument events, alarms and errors. Each event has a Type which describes the category of the event, and an ID which describes the actual event. Events are ordered with the most recent being at Event1 and the least recent at Event40.

Status Word to Indicate Instrument Errors Via Comms

The bits in the status word indicate the following errors:

Bit No.	Description
0	At least 1 Fatal error has occurred
1	At least 1 Config error has occurred
2	At least 1 HW MissMatch error has occurred
3	At least 1 power module ribbon cable error, eeprom error, revision error has occurred
4	At least 1 Power module is running using default calibration
5	At least 1 option IO module or the standard IO is using default calibration
6	One or more network tasks is running in Not Firing Mode
7	Reserved

Configuration Mode Parameters

To write parameters in this group, it is first necessary to set the 'Access.IM' parameter (Modbus 199 – 00C7_{hex}) to the value 2 to set the controller into configuration mode. Note this will disable all normal control action and the controller outputs will be switched to a safe state.

It is not necessary to set any 'password' parameters to enter configuration mode.

To exit from configuration mode, simply write 0 to instrument mode. This will reset the controller, a process that takes several seconds. During this period it will not be possible to communicate with the controller.

It is also possible to write the value 1 to the 'Access.IM' parameter to put the EPower controller unit into standby mode.

Modbus Advanced Topics

Access to Full Resolution Floating Point and Timing Data

One of the main limitations of Modbus is that only 16 bit integer representations of data can normally be transferred. In most cases, this does not cause a problem, since appropriate scaling can be applied to the values without losing precision. Indeed all values displayable on the 4 digit EPower controller front panel may be transferred in this way. However, this has the significant drawback that the scaling factor to be applied needs to be known at both ends of the communications link.

One further problem is that certain 'time' parameters, are always returned over the communications link in either 10^{ths} of seconds or 10^{ths} of minutes, configured via Instrument.Configuration.TimerRes. It is possible for long durations to overflow the 16 bit Modbus limit.

To overcome these problems, a sub protocol has been defined, using the upper portion of the Modbus address space (8000h and upwards), allowing full 32 bit resolution floating point and timer parameters. The upper area is known as the IEEE region.

This sub-protocol provides two consecutive Modbus addresses for all parameters. The base address for any given parameter in the IEEE region can easily be calculated by taking its normal Modbus address, doubling it, and adding 8000h. For example, the address in the IEEE region of the Target Setpoint (Modbus address 2) is simply:

$$2 \times 2 + 8000h = 8004h = 32772 \text{ decimal}$$

This calculation applies to any parameter that has a Modbus address.

Access to the IEEE area is made via block reads (Functions 3 & 4) and writes (Function 16). Attempts to use the 'Write a Word' (Function 6) operation will be rejected with an error response. Furthermore, block reads and writes using the IEEE region should only be performed at even addresses, although no damage to the instrument will result in attempting access at odd addresses. In general, the 'number of words' field, in the Modbus frame, should be set to 2 times what it would have been for 'normal' Modbus.

The rules governing how the data in the two consecutive Modbus addresses are organised depending on the 'data type' of the parameter.

Data Types Used in EPower Controller Units

See PARAMETER TABLE in section 8 of the EPower controller User Guide Part No. HA179769.

- Enumerated parameters are parameters which have a textual representation for their value on the user interface, for example, 'Parameter Status' – 'Good/Bad', 'Analog Operator Type' – 'Add', 'Subtract', 'Multiply', etc.
- Booleans are parameters which can have either a value '0' or a value '1'. Generally these parameters are enumerated. These are denoted as 'bool' in the table.
- Status words are generally only available over communications, and are used to group binary status information. See "Status Words" on page 41.

- Integer parameters are those that never include a decimal point however the instrument is configured, and do not refer to a time period or duration. These include such values as the instrument communications address and values used to set passwords, but not Process Variable and Setpoint related parameters, even if the display resolution of the instrument is set to no decimal places. These may be 8 or 16 bit and are denoted by 'uint8' or 'uint16' unsigned integers or 'int8' or 'int16' signed (+ or -) integers.
- Floating point parameters are those having a decimal point (or those which may be configured to have a decimal point), with the exception of parameters relating to time periods and duration. This includes Process Variable, Setpoints, Alarm Setpoints, etc and are denoted as type 'Float32' (IEEE 32-bit floating point parameters).
- Time Type parameters measure durations, for example, Alarm time above threshold, Timer elapsed time, etc. These are denoted by 'time32' in the parameter table.

Enumerated, Status Word, and Integer Parameters

These use only the first word of the 2 Modbus addresses assigned to them in the IEEE area. The second word is padded with a value of 8000 hex.

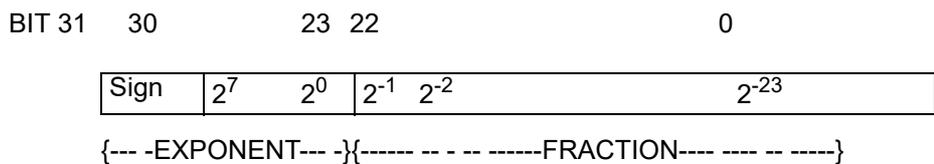
Although 'Write a Word' (Function 6) is not permitted, this type of parameter may be written as a single 16 bit word using a Modbus 'Block Write' (Function 16). It is not necessary to add a padding value in the second address. Similarly, such parameters may be read using a Modbus 'Block Read' (Function 3 & 4) as single words, in which case the padding word will be omitted.

It is, however, necessary to pad the unused word when writing this sort of data types as part of a block containing other parameter values.

Floating Point Parameters

These use the IEEE format for floating point numbers, which is a 32 bit quantity. This is stored in consecutive Modbus addresses. When reading and writing to floats, it is necessary to read or write both words in a single block read or write. It is not possible, for example, to combine the results of two single word reads.

This format is used by most high level programming languages such as 'C' and BASIC, and many SCADA and instrumentation systems allow numbers stored in this format to be decoded automatically. The format is as follows:



where value = $(-1)^{\text{Sign}} \times 1.F \times 2^{E-127}$

Note that in practice, when using C, IEEE floats may usually be decoded by placing the values returned over comms into memory and ‘casting’ the region as a float, although some compilers may require that the area be byte swapped high to low before casting. Details of this operation are beyond the scope of this manual.

The format used to transfer the IEEE number is as follows:

Lower Modbus Address		Higher Modbus Address	
MSB	LSB	MSB	LSB
Bits 31 - 24	Bits 16 - 23	Bits 15 - 8	Bits 7 - 0

For example, to transfer the value 1.001, the following values are transmitted (hexadecimal).

Lower Modbus Address		Higher Modbus Address	
MSB	LSB	MSB	LSB
3F	80	20	C5

Time Type Parameters

Time type values are returned over comms in 1/10 seconds or minutes. This can be changed in the SCADA table. Time durations are represented as a 32 bit integer number of milliseconds in the IEEE area. When reading and writing to time types, it is necessary to read or write both words in a single block read or write. It is not possible, for example, to combine the results of two single word reads.

The data representation is as follows:

Lower Modbus Address		Higher Modbus Address	
MSB	LSB	MSB	LSB
Bits 31 - 24	Bits 16 - 23	Bits 15 - 8	Bits 7 - 0

To create a 32 bit integer value from the two Modbus values, simply multiply the value at the lower Modbus address by 65536, and add the value at the Higher address. Then divide by 1000 to obtain a value in seconds, 60000 for a value in minutes, etc.

For example, the value of 2 minutes (120000 mS) is represented as follows:

Lower Modbus Address		Higher Modbus Address	
MSB	LSB	MSB	LSB
00	01	D4	C0

Programmable Logic Controllers and EPower Controller

There are many ways of connecting EPower controller units to Programmable Logic Controllers using Modbus. It is usually best to avoid the use of Basic modules which may result in very slow communications. Your supplier will often be able to advise on a solution for a particular make of Programmable Logic Controller, but if requesting information from third party vendors, note that the EPower controller units support standard Modbus RTU, allowing use of function 16 for block write operations, and functions 3 and 4 for reads.

Ethernet (Modbus TCP), Single Port

Overview

EPower controller units support the Modbus/TCP protocol using Ethernet. This protocol, when used over Modbus/TCP single port hardware, embeds the standard Modbus protocol within an Ethernet TCP layer.

From V3.10, EPower controller units support the Modbus/TCP protocol over dual port hardware. It embeds the standard Modbus protocol but with some limitations, see "Ethernet (Modbus TCP), Single Port".

Note: The following information in this chapter is only applicable to Modbus/TCP single port hardware.

As most parameters are saved in the EPower controller unit's memory, the interface board must retrieve these values before it can start communicating on Ethernet.

The values seen by EPower controller via the configuration cable in this case will vary according to the instrument set-up:

1. The MAC (Media Access Control) address will be returned as 0 in all fields.
2. If DHCP is configured, the IP address, Subnet mask and Default gateway will contain the values last assigned by the DHCP server. These may change when the configuration cable is removed.

Changing any of the IP parameters will cause the interface board to reset in order to retrieve the new values.

Any socket left with no data traffic for 2 minutes will be disconnected and made available for new connections.

Support for other Ethernet Utilities

In addition to the MODBUS TCP protocol, the units support the standard Ethernet 'ping' utility to assist in fault finding. Other interfaces such as HTTP, FTP or TelNet are not currently supported.

Ethernet Wiring

Ethernet capability is provided by an interface board installed within the unit and provides a single RJ45 socket ("Modbus TCP (Ethernet 10baseT), Single Port" on page 17). This interface board communicates to the unit internally using a standard Modbus interface.

The Ethernet port is a 10baseT port and can be connected to a hub or switch with Cat5 cable via the standard RJ45 connector. Alternatively, an RJ45 cross-over cable may be used to connect direct to a PC 10baseT Network Interface Card. The maximum cable length for 10baseT is 100 metres. 10baseT operates at 10 Mbps and uses baseband transmission methods.

The interconnecting cables should be fitted with plugs provided with an outer metallic shell with the shell connected to the wire screen of the cable. This type of cable must be used to maintain EMC compliance.

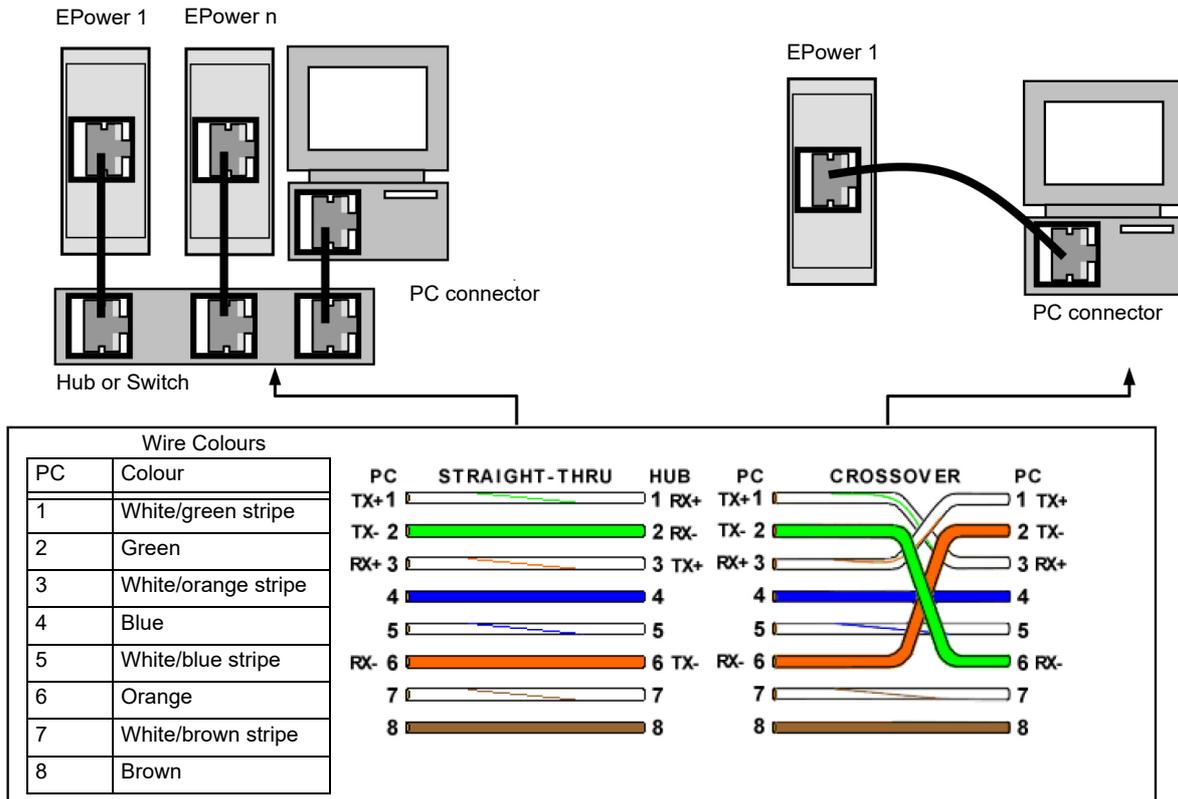


Figure 22 Ethernet (Modbus TCP) Wiring - Single and Multiple EPower controller Units

Instrument Setup

It is recommended that the communications settings for each instrument are set up *before connecting it to any Ethernet network*. This is not essential but network conflicts may occur if the default settings interfere with equipment already on the network.

For normal Modbus (and other protocols) there is only the one address parameter that needs to be set. For the Ethernet instruments, however, there are several more: IP address, subnet mask, default gateway and DHCP enable. These parameters are available in EPower controller units under different levels of access as detailed in the EPower Controller User Guide (HA179769).

Changing any one of these parameters may immediately move the instrument to a new network address. For this reason, it is recommended that such changes are made offline.

IP Addresses are usually presented in the form "abc.def.ghi.jkl". In the instrument Comms folder each element of the IP Address is shown and configured separately such that IPAdd1 = abc, IPAddr2 = def, IPAddr3 = ghi and IPAdr4 = jkl.

This also applies to the SubNet Mask, Default Gateway and Preferred Master IP Address.

Each Ethernet module contains a unique MAC address, normally presented as a 12 digit hexadecimal number in the format "aa-bb-cc-dd-ee-ff".

In EPower controller units MAC addresses are shown as 6 separate decimal values in iTools. MAC1 shows the first pair of digits in decimal, MAC2 shows the second pair of digits and so on.

Unit Identity

The Modbus TCP Specification includes the 'normal' Modbus address as part of the packaged Modbus message – where it is called the Unit Identifier. If such a message is sent to an Ethernet / Serial gateway, the UnitID is essential to identify the slave instrument on the serial port. When a stand alone Ethernet instrument is addressed, however, the UnitID is surplus to requirements since the IP address fully identifies the instrument. To allow for both situations the UnitID Enable parameter is used to enable or disable checking of the UnitID received from TCP. The enumerations produce the following actions:

'Instr'	the received Unit Ident must match the Modbus address in the instrument or there will be no response.
'Loose'	the received Unit Ident value is ignored, thus causing a reply regardless of the received Unit Ident.
'Strict'	the received Unit Ident value must be 0xFF or there will be no reply.

Dynamic Host Configuration Protocol (DHCP) Settings

This is set in configuration level by the DHCP Enable parameter.

IP addresses may be 'fixed' – set by the user, or dynamically allocated by a DHCP server on the network.

If IP Addresses are to be dynamically allocated the server uses the instrument MAC address to uniquely identify them.

Fixed IP Addressing

In the "Comms" folder of the instrument set the "DHCP enable" parameter to "Fixed". Set the IP address and SubNet Mask as required. This may be done in Engineer level.

Dynamic IP Addressing

In the "Comms" folder of the instrument set the "DHCP enable" parameter to "Dynamic". Once connected to the network and powered, the instrument will acquire its "IP address", "SubNet Mask" and "Default Gateway" from the DHCP Server and display this information within a few seconds.

Note that if the DHCP server does not respond (in common with other Ethernet appliances in this situation) the unit will not be accessible via the network.

Default Gateway

The "Comms" folder also includes configuration settings for "Default Gateway", these parameters will be set automatically when Dynamic IP Addressing is used. When fixed IP addressing is used these settings are only required if the instrument needs to communicate wider than the local area network i.e. over the internet.

Preferred Master

The "Comms" folder also includes configuration settings for "Preferred Master". Setting this address to the IP Address of a particular PC will guarantee that one of the available Ethernet sockets will always be reserved for that PC.

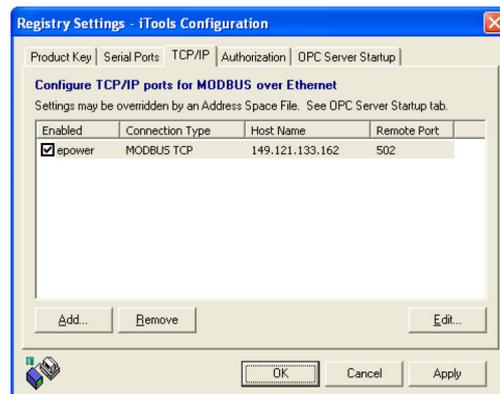
iTools Setup

iTools configuration package, version V7 or later, may be used to configure Ethernet communications.

The following instructions configure Ethernet.

To include a Host Name/Address within the iTools scan:

1. Ensure iTools is NOT running before taking the following steps
2. Within Windows, select 'Control Panel'
3. In control panel select 'iTools'
4. Within the iTools configuration settings select the 'TCP/IP' tab
5. Click the 'Add' button to add a new connection
6. Enter a name for this TCP/IP connection
7. Click the 'Add' button to add the host name or IP address of the instrument in the 'Host Name/ Address' section
8. Click 'OK' to confirm the new Host Name/IP Address you have entered
9. Click 'OK' to confirm the new TCP/IP port you have entered
10. You should now see the TCT/IP port you have configured within the TCP/IP tab of the iTools control panel settings



iTools is now ready to communicate with an instrument at the Host Name/IP Address you have configured.

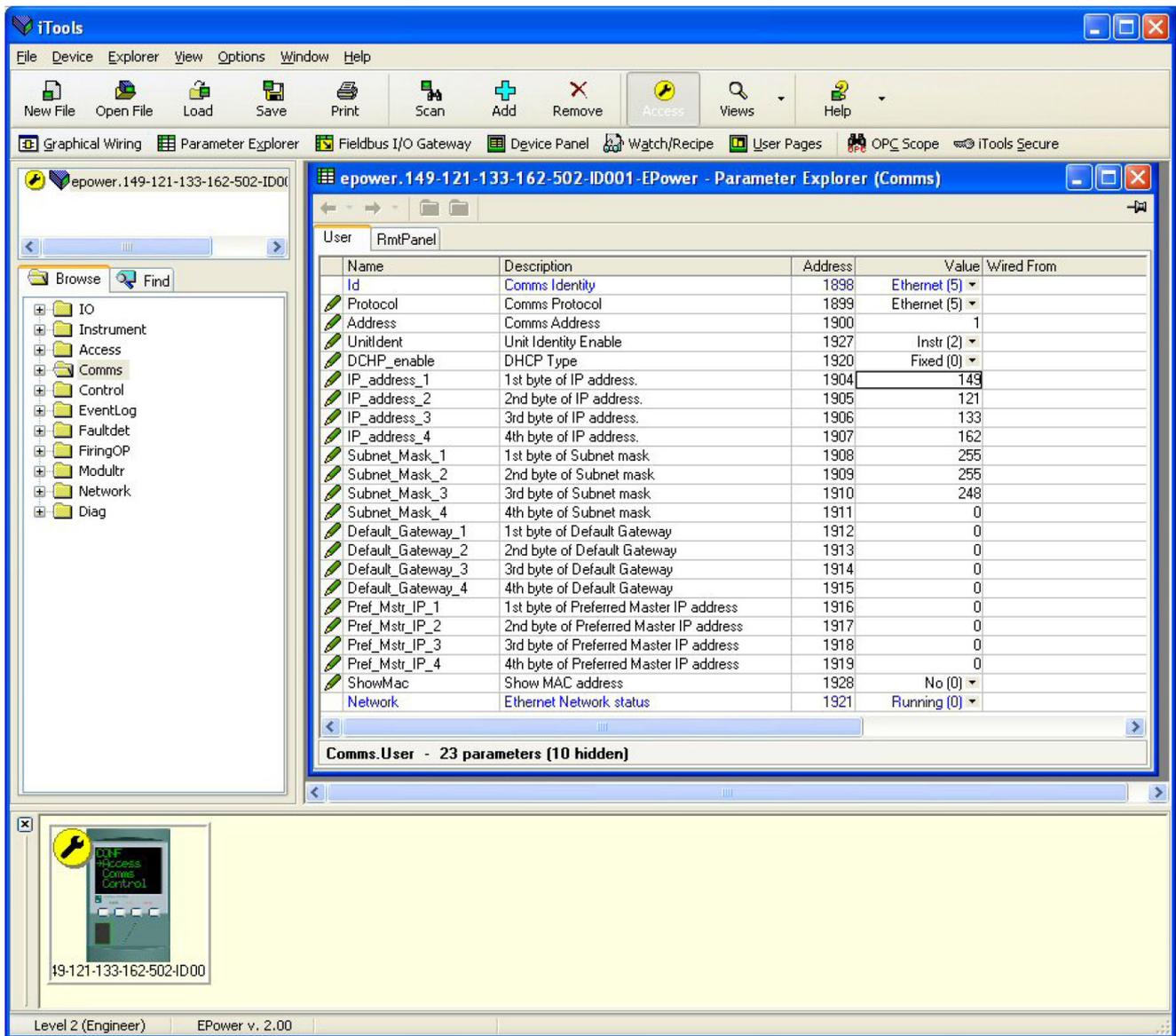


Figure 23 Ethernet Communications Parameters

Ethernet (Modbus TCP), Dual Port

Overview

EPower controller units support the Modbus/TCP through Ethernet dual port provided by a dual RJ45 socket ("Modbus TCP (Dual Port)" on page 18). It embeds the standard Modbus protocol within an Ethernet TCP layer but with some differences compared to description in "Modbus and Jbus Protocol", regarding function codes and parameter address (see "Function Codes and Address Space" on page 55).

As most parameters are saved in the EPower controller unit's memory, the interface board must retrieve these values before it can start communicating on Ethernet.

The values seen by EPower controller via the configuration cable in this case will vary according to the instrument set-up:

1. The MAC (Media Access Control) address will be returned as 0 in all fields.
2. If DHCP is configured, the IP address, Subnet mask and Default gateway will contain the values last assigned by the DHCP server. These may change when the configuration cable is removed.

Changing any of the IP parameters will cause the interface board to reset in order to retrieve the new values. Any socket left with no data traffic for 2 minutes will be disconnected and made available for new connections.

Function Codes and Address Space

The following Modbus function are implemented in the module:

Function code	Function	Address space access
03	Read n words	FieldBus IO Gateway only
06	Write a word	FieldBus IO Gateway only
16	Write n words	FieldBus IO Gateway only
23	Read/Write n words	Whole EPower address space

Function codes 03, 06 and 16 can only be used to access Fieldbus IO Gateway parameters.

Function code 23 can be used to access the complete EPower address map as described in "Parameter Address" on page 30.

Inputs	Address
Input01	0x0100
Input02	0x0101
Input03	0x0102
Input04	0x0103
Input05	0x0104
Input06	0x0105
Input07	0x0106
Input08	0x0107
Input09	0x0108
Input10	0x0109
Input11	0x010A
Input12	0x010B
Input13	0x010C
Input14	0x010D
Input15	0x010E
Input16	0x010F
Input17	0x0110
Input18	0x0111
Input19	0x0112
Input20	0x0113
Input21	0x0114
Input22	0x0115
Input23	0x0116
Input24	0x0117
Input25	0x0118
Input26	0x0119
Input27	0x011A
Input28	0x011B
Input29	0x011C
Input30	0x011D
Input31	0x011E
Input32	0x011F

Outputs	Address
Output01	0x0000
Output02	0x0001
Output03	0x0002
Output04	0x0003
Output05	0x0004
Output06	0x0005
Output07	0x0006
Output08	0x0007
Output09	0x0008
Output10	0x0009
Output11	0x000A
Output12	0x000B
Output13	0x000C
Output14	0x000D
Output15	0x000E
Output16	0x000F

Table 1: Modbus TCP dual port, Fieldbus IO Gateway parameters

Support for other Ethernet Utilities

In addition to the MODBUS TCP protocol, the units support the standard Ethernet ‘ping’ utility to assist in fault finding. Other interfaces such as HTTP, FTP or TelNet are not currently supported.

Ethernet Wiring

Ethernet capability is provided by an interface board installed within the unit and provides a dual RJ45 sockets ("Modbus TCP (Dual Port)" on page 18). The Ethernet interface supports 10/100 Mbit/s, full or half duplex operation.

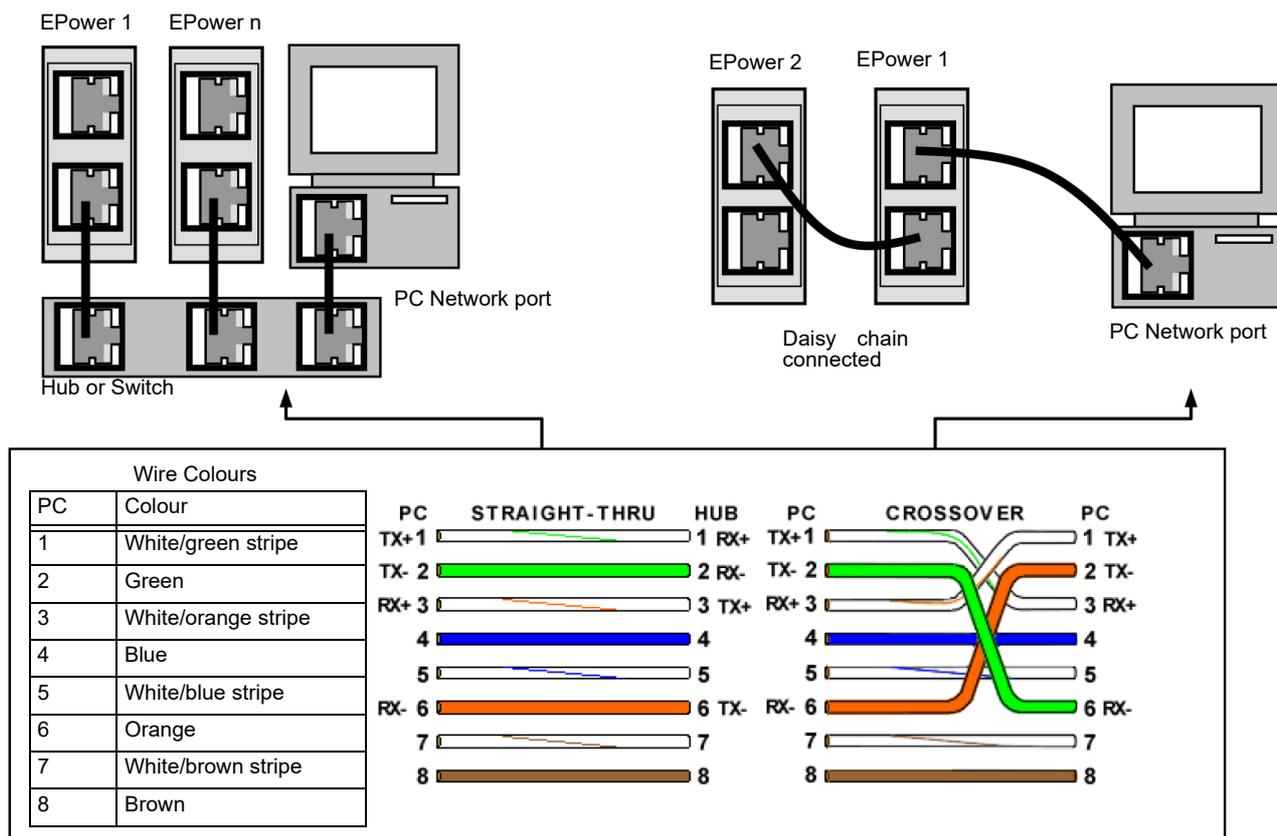


Figure 24 Ethernet (Modbus TCP), dual port – Wiring, Single and Multiple EPower controller Units

Instrument Setup

See "Instrument Setup" on page 57.

Tools Setup

ModbusTCP dual port does not support iTools connection.

Profibus

Introduction

Profibus DP is an industry standard open network used to interconnect instrumentation and control devices in, for example, a manufacturing or processing plant. It is often used to allow a central Programmable Logic Controller (PLC) or PC based control system to use external 'slave' devices for input/output (I/O) or specialised functions, thus reducing the processing load on the controlling unit so that its other functions can be carried out more efficiently using less memory.

The Profibus network use a high speed version of the EIA485 standard (see also "Wiring General" on page 15), and permits transmission rates of up to 12M Baud between the host and up to 32 Profibus 'Stations' or 'nodes' within a single section of a network. The use of repeaters, such as KD485 – "Interconnection Diagram 3-Wire EIA485" on page 23, (each counted as a node) allows the maximum of 127 nodes (addresses 0 to 127) to be supported.

Profibus DP distinguishes between master and slave devices. It allows slave devices to be connected on a single bus thus eliminating considerable plant wiring.

Master devices determine the data communications on the bus. A master can send messages without an external request when it holds the bus access rights (the token). Masters are also called active stations in the Profibus protocol.

Slave devices are peripheral devices such as I/O modules, valves, temperature controllers/indicators, and measuring transmitters. EPower controller units are intelligent slaves which will only respond to a master when requested to do so.

Profibus DP is based around the idea of 'cyclical scan' of devices on the network, during which 'input' and 'output' data for each device is exchanged.

It is not within the scope of this document to describe the Profibus standard in detail. This may be found by reference to www.profibus.com. Similarly, further details of EPower controllers may be found by reference to the EPower controller User Guide Part No. HA17969.

The following general points apply to EPower controller units.

- Profibus slave communications is provided in EPower controller units by means of a plug in module. The connection to Profibus is via a standard 9-way D connector, see section 0.
- Baud rates of up to 12MB are provided. The baud rate is automatically detected by the EPower controller unit.
- Profibus DP Cyclic data transfer is provided, as well as DPV1 C1 and C2 Acyclic communications.
- A maximum of 16 input words and 16 output words may be transferred between EPower controller and the Profibus master using the Profibus cyclic I/O. These may be chosen from any of the wireable parameters in the unit.
- EPower controller differs from previous Eurotherm Profibus slave devices in that it is not configured via the GSD file editor. Instead, the I/O data mappings are set up using iTools, and there is a single, fixed, GSD file.

- The 'Demand Data' protocol used previously is not supported. Instead DPV1 Acyclic communications are provided to give access to variables not included in the Cyclic I/O definitions.
- All variables are returned as 16 bit signed 'scaled' integers, such that 999.9 is returned or sent as 9999; 12.34 is encoded as 1234. The control program in the Profibus master must convert the numbers into floating point values if required. For example a power setpoint of 50.0% is encoded as an integer value of 500.

Wiring General

The wiring principles described in "Wiring General" on page 15 apply to Profibus. Line termination is different from standard EIA485 as shown in [Figure 25](#).

The Profibus cable ("Cable Selection" on page 61) is a single cable running through the plant with, usually, the Profibus master at one end. Nodes can be connected at convenient points along the cable as shown in principle below. It is recommended that the cable length between a node and the Profibus cable should not exceed 2m. The connection to a node is normally via a 9-way connector assembly of the type which allows the cables to be connected using screw or spring-cage terminal blocks. These 9-pin connectors are easily available for Profibus networks. Terminating resistors may be integrated in the assembly and enabled externally using a slide switch or they may be wired in. Make sure, however, that the terminating resistors are only in circuit on the last node in the chain.

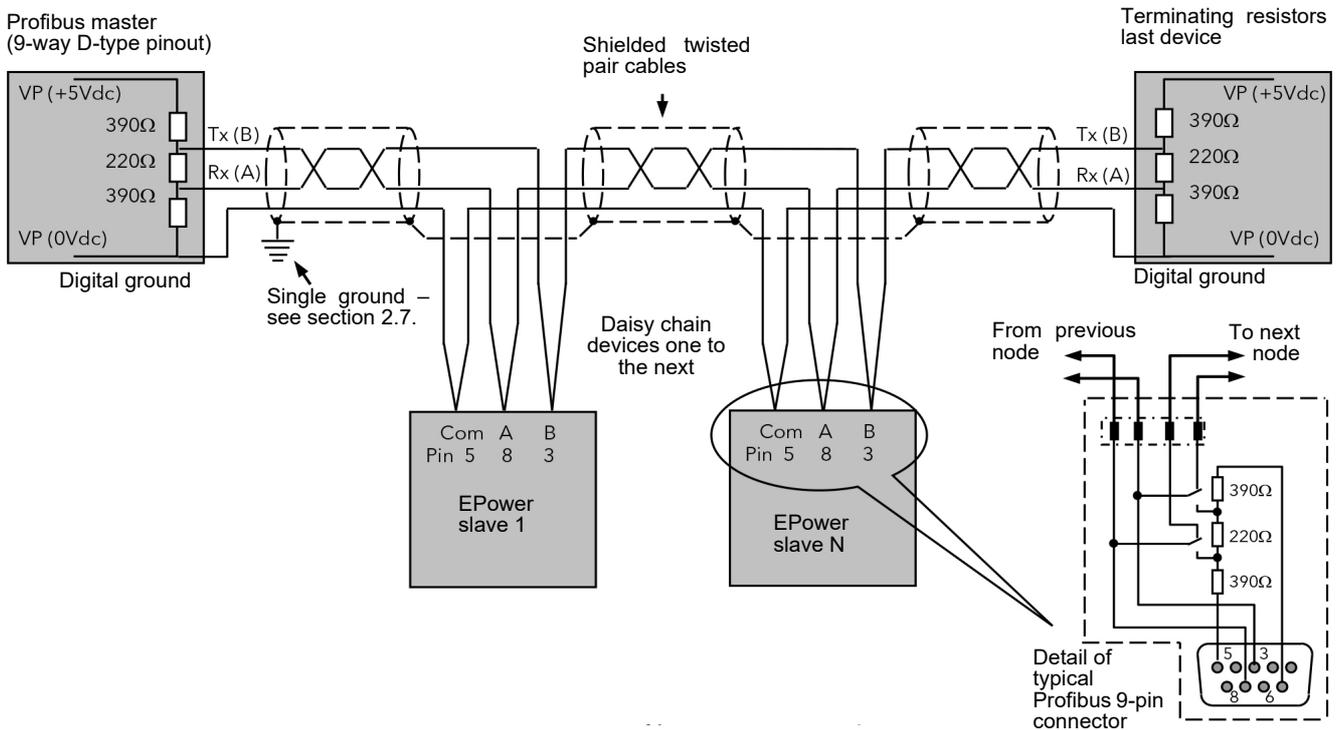


Figure 25 Profibus Wiring General

Cable Selection

The table below gives the specification for a suitable cable such as Beldon B3079A.

Impedance	135 to 165 ohms at 3 to 20 MHz
Resistance	<110 ohms/km
Cable capacitance	<30pF/metre
Core diameter	>0.34 mm ² (22awg)
Cable type	Twisted pair, 1x1, 2x2 or 4x1 lines
Signal attenuation	9 dB max. over total length of line section
Shielding	Cu shielding braid, or shielding braid and shielding foil

Maximum Baud Rate Compared with Cable Length

The maximum transmission speed depends on the length of the cable run including 'stub' (distance from the bus to a station) lengths. Recommended minimum values are:

Line length/segment (metres)	100	200	400	1000	1200
Maximum baud rate (kbit/sec) kB	12,000	1,500	500	187.5	93.75

Node Address

Each node must be given a unique address. This may be done in iTools or the EPower controller user interface. The parameter is called 'Address' which may be found in the 'Comms' list and may be changed in Engineer level. This is further described in the EPower controller User Guide. The unit is shipped with a default address of 1. This is within the address range of the Profibus protocol (0 to 126), so if the unit is inadvertently inserted into the network without a new address having been set, the bus may be affected.

Note: After changing the Profibus address, the EPower controller unit should be powered off and on again, to allow correct initialisation to take place.

To set the address using iTools, open the Comms list and double click the 'User' sub-folder to open the list of parameters. Enter the value for the Comms Address.

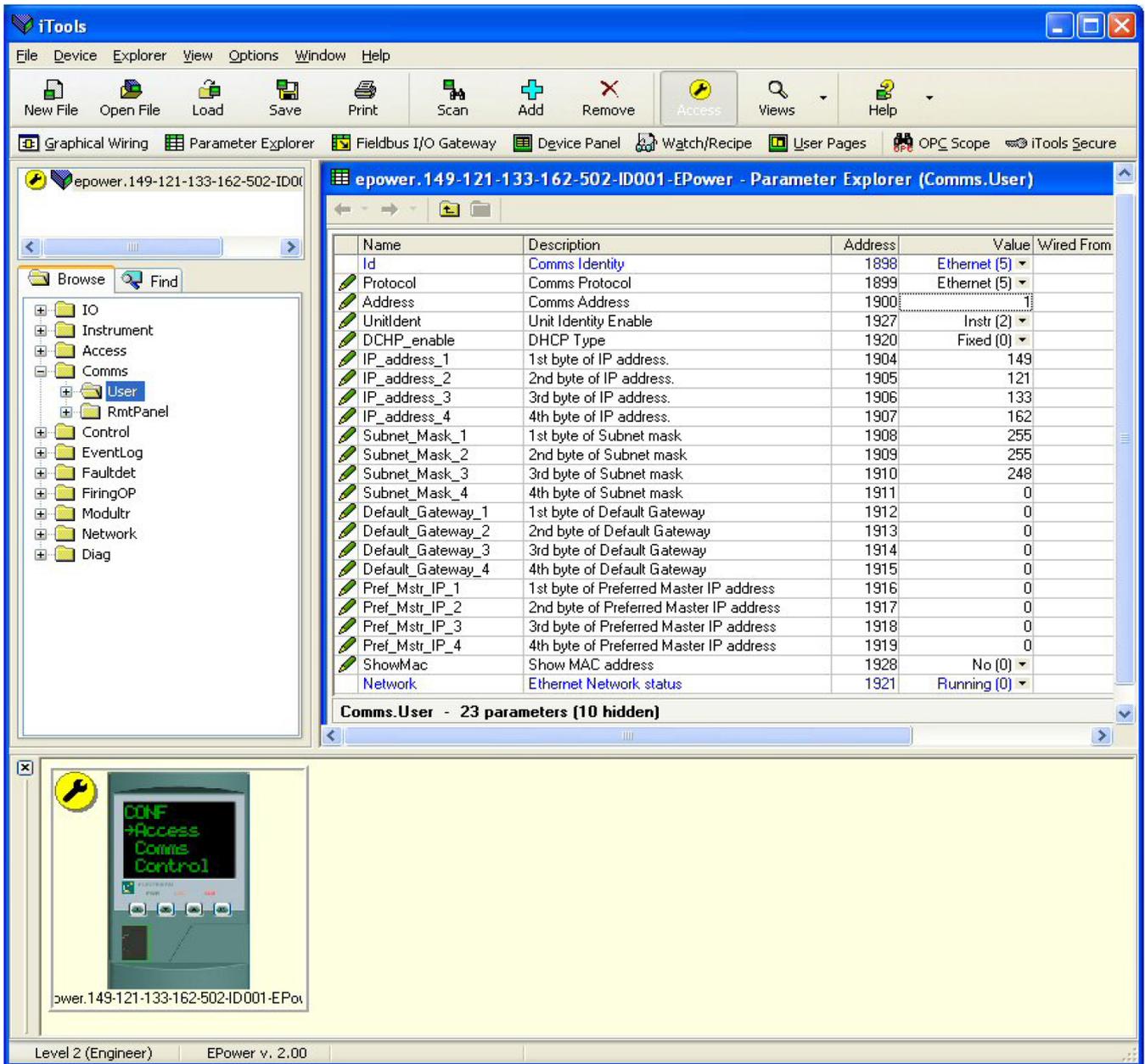


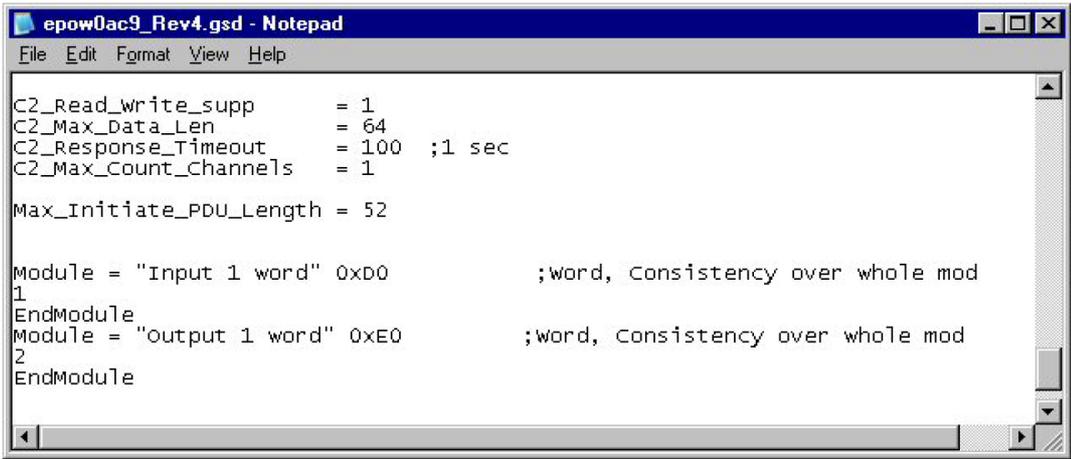
Figure 26 Setting the Profibus Comms Address in iTools

Configuring the Data Exchange

The Profibus network master (typically a PLC) may be required to work with many diverse slaves from different manufacturers and with different functions. EPower controller units contain many parameters most of which will not be required in Cyclic Data Exchange by the network master for a particular application. It is, therefore, necessary for the user to define which Input and Output parameters are to be available for Cyclic Data Exchange on the Profibus link. Note that all parameters are available in Acyclic Data Exchange. The master may then map the selected device parameters into the PLC input/output registers, or, in the case of a supervisory (SCADA) package, to a personal computer.

Previous Eurotherm instruments have used the 'Eurotherm GSD Editor' to define input and output data. This produces a text file in which the individual input and output parameters are listed. For EPower, however, the process is simplified by making use of the 'Fieldbus I/O Gateway Editor' in iTools to create the assignment. The programming interface consists of defining individual modules which hold the list of parameters that have to be exchanged. Each module holds a single parameter and there are two types of module 'Input 1 Word' and 'Output 1 Word', as shown in the text file in [Figure 27](#).

The Eurotherm GSD Editor, therefore, cannot be used when setting up EPower for Profibus.



```
epow0ac9_Rev4.gsd - Notepad
File Edit Format View Help
C2_Read_write_supp      = 1
C2_Max_Data_Len         = 64
C2_Response_Timeout     = 100 ;1 sec
C2_Max_Count_Channels   = 1

Max_Initiate_PDU_Length = 52

Module = "Input 1 word" 0xD0      ;word, Consistency over whole mod
1
EndModule
Module = "Output 1 word" 0xE0     ;word, Consistency over whole mod
2
EndModule
```

The GSD file supplied is never changed and is imported, as supplied, into the master device.

Figure 27 Text file showing two pre-defined module types in the EPower GSD file

Values from each slave, 'Input Data', are read by the master, which then uses the values in its own application, (such as a ladder logic program). The master also generates a set of values, 'Output Data', into a pre-defined set of registers to be transmitted to the slaves. This process is called an 'I/O data exchange' and is repeated continuously, to give a cyclical I/O data exchange.

Select the 'Fieldbus I/O Gateway' tool from the lower toolbar, and an editor screen will appear similar to the picture below.

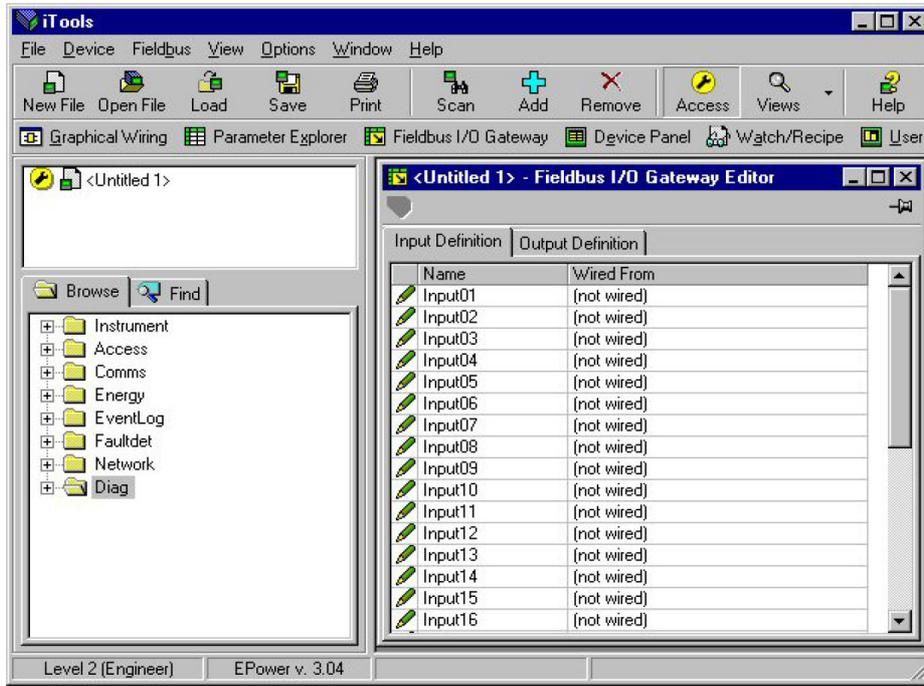


Figure 28 The Fieldbus I/O Gateway Editor in iTools

There are two tabs within the editor, one for the definition of Inputs, and the other for Outputs. 'Inputs' are values sent from the EPower controller to the Profibus master, for example, alarm status information or measured values, i.e. they are readable values. 'Outputs' are values received from the master and used by the EPower controller, for example, setpoints written from the master to EPower controller. Note that Outputs are written on every Profibus cycle, which is frequent, of the order of 10-100mS, and so values from Profibus will overwrite any changes made on the EPower controller keypad unless special measures are taken to prevent this.

The procedure for selecting variables is the same for both input and output tabs. Double click the next available position in the input or output data and select the variable to assign to it. A pop-up provides a browser from which the required parameter can be obtained. Double click the parameter to assign it to the input definition and repeat this for all required parameters. Note that you should assign inputs and outputs contiguously, as a 'not wired' entry will terminate the list even if there are assignments following it. [Figure 29](#) shows an example of the pop-up and the input list produced.

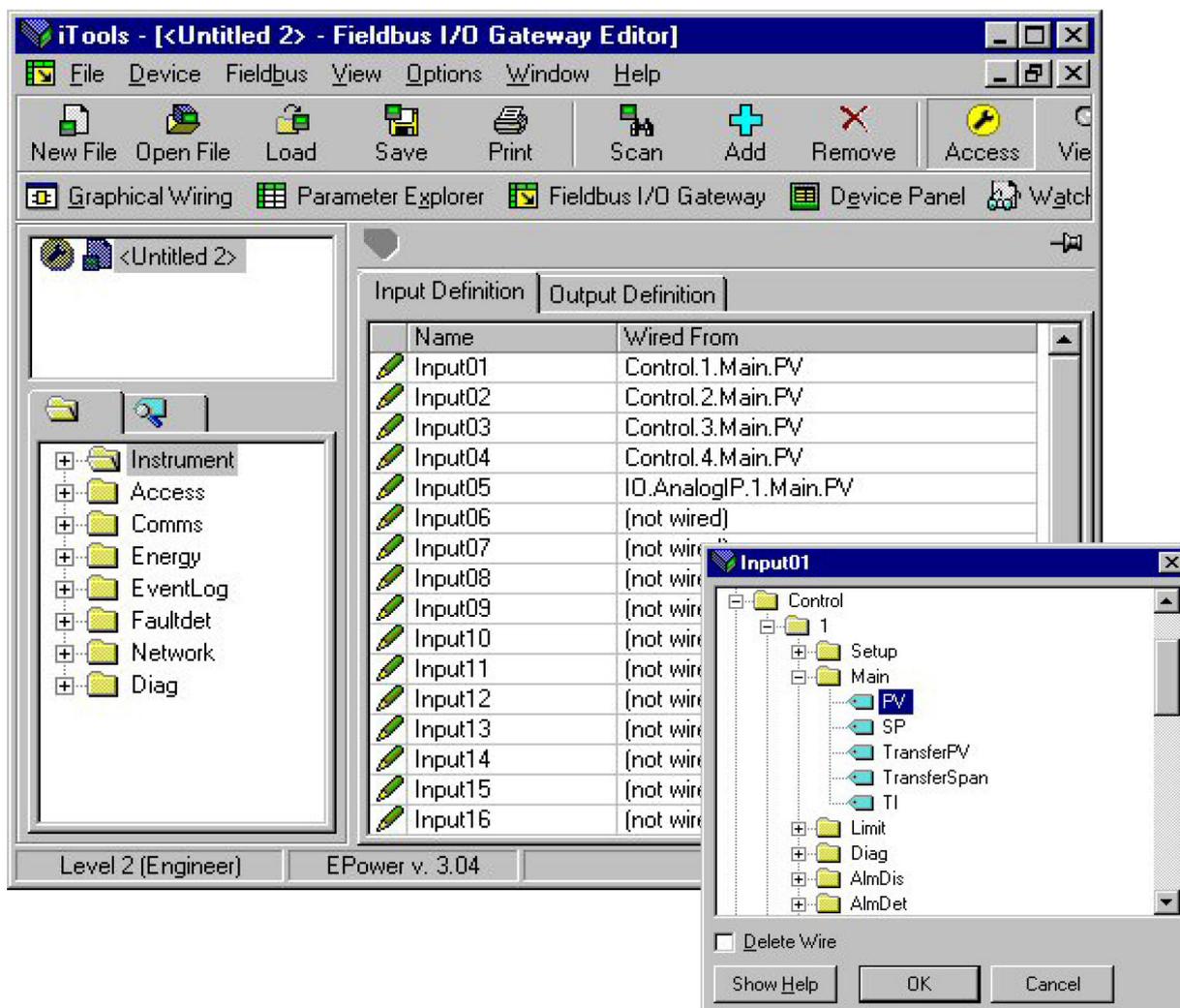


Figure 29 Selecting an Input Value and Example of an Input List

The Profibus standard allows a maximum of 117 total input and output parameters but most Profibus masters are unable to deal with this number. A maximum of 32 input and 16 output parameters may be set using the Fieldbus I/O Gateway Editor.

When the list is populated with the variables you require, note how many ‘wired’ entries are included in each of the input and output areas as this will be needed when setting up the Profibus Master.

Notes:

1. No checks are made that output variables are writeable, and if a read only variable is included the output list any values sent to it over Profibus cyclic communications will be ignored with no error indication.
2. Following a Quick Start, four parameters are pre-loaded by default in the Input buffer and four in the Output buffer. These are:

Input 01	Control.1Main.PV	Output 01	SetPwr.1.Remote.1
Input 02	Control.2Main.PV	Output 02	SetPwr.2.Remote.1
Input 03	Control.3Main.PV	Output 03	SetPwr.3.Remote.1
Input 04	Control.4Main.PV	Output 04	SetPwr.4.Remote.1

These parameters represent commonly used parameters. They can be removed if they are not required in a particular application but they must be replaced contiguously.

Once the changes have been made to the I/O lists, they must be downloaded to the EPower controller unit.

This is done with the button on the top left of the I/O Editor marked . The EPower controller Unit will need to be powered off and on again once this has been done for the changes to register.

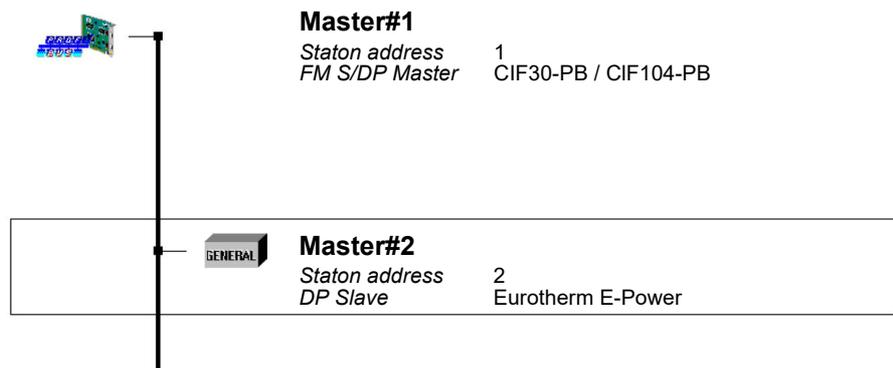
The next step in the process is to configure the Profibus master.

To Configure the Profibus Master

In order to configure Profibus networks, GSD files (<name>.GSD) must be imported into the network configuration tool provided by the vendor of the Profibus master device. Typically this may be a PLC using a 'PLC Configurator' tool. The GSD file is a text file in a predefined standardised format. It is used to declare a new slave to the master and to allow communications between the master and the slave. The required GSD file for EPower controller units is available on the support disc packaged with the firmware upgrade tool.

Only general indications on how this is done can be provided, since this is Profibus Master specific. However, the general procedure will be as follows:

1. Import the GSD file for EPower controller, named EPOW0AC9.GSD, into your master configuration tool.
2. Create a slave node based on this GSD file and assign a Slave Address. This is usually done by double clicking on the slave icon on a graphical network representation provided by the master configuration tool.
3. Assign modules to the slave. There are two modules which are already pre-defined in the GSD file for EPower, These are "Input 1 word", and "Output 1 word". In the master you must declare the number of input modules and the number of output modules for every item in the input and output lists, with the input lists being assigned first. So, for the example given in the previous section, (see [Figure 30](#)) you should add the "Input 1 word" five times to the module list, and then add sufficient "Output 1 word" modules to cover the number of outputs you have defined. Unless this step is performed correctly, Profibus communications will not be established. [Figure 30](#) shows an example of how this may be applied in a typical master for one slave at address 2. In this example, five input words and two output words are defined.



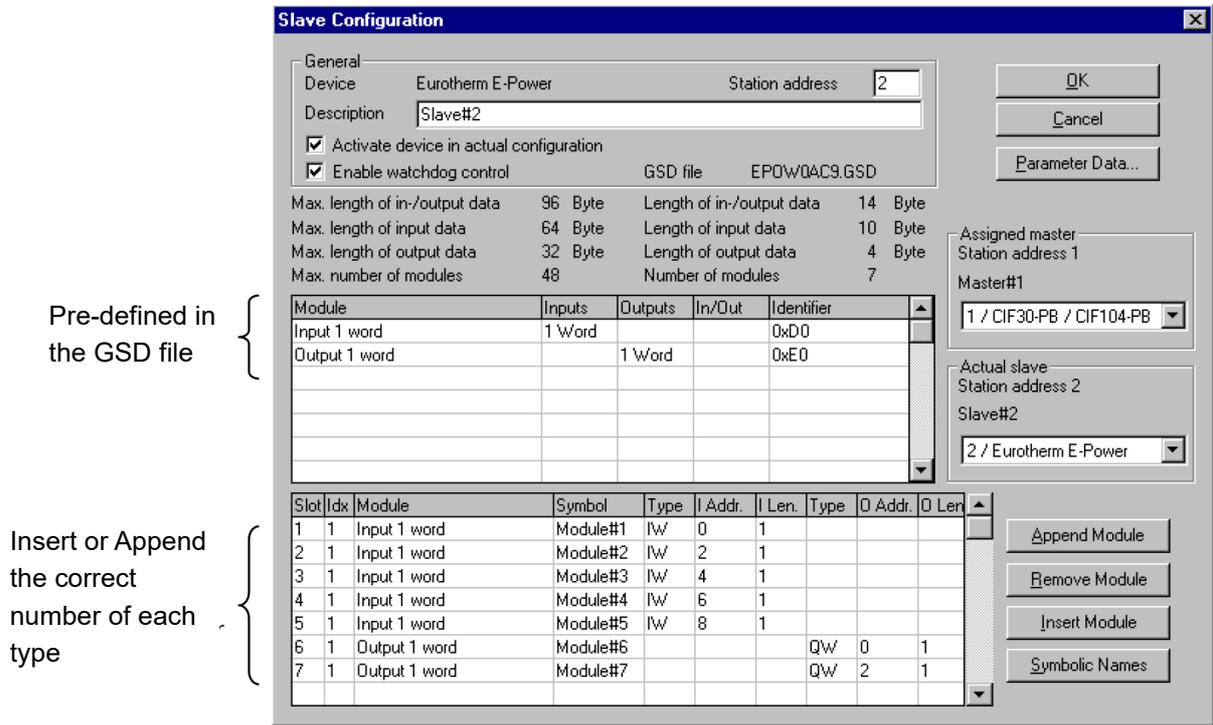


Figure 30 Configuration example in a typical mater

- Save and download the master configuration, and put the master online. The top LED on the Profibus interface (above the D Connector) should light solid green showing that communication is established.
- If the light flashes, does not turn on, or lights as red, verify the proceeding steps have been correctly performed.

DPV1 Acyclic Communications

DPV1 C1 and C2 Acyclic communications are provided so that lesser used EPower controller parameters may be read and written when needed. The method used to use DPV1 is master specific and not described further in this document, however blocks of up to 32 variables (64 bytes) may be returned or written in a single transaction.

All EPower controller parameters and variables may be accessed. The Slot and Index value to be used for a parameter is derived from the tag address for the parameter (numerically equal to the Modbus address) using the following calculation:

$$\text{Slot} = (\text{tag} - 1) / 255$$

$$\text{Index} = (\text{tag} - 1) \text{ MOD } 255$$

Trouble-Shooting

No Communications

- Check the wiring carefully, verifying the continuity of the A and B connections to the master. Ensure correct terminals have been used.
- Check the node address is correct for the network configuration in use. Ensure the address is unique.
- Ensure that the network has been correctly configured and that the configuration has been correctly downloaded to the master.
- Verify that the GSD file being used is correct.

5. Ensure that the maximum line length of the transmission line has not been exceeded for the baud rate in use.
6. Ensure that the final node on the transmission line (no matter what type of instrument it is) is terminated correctly using three resistors as shown in [Figure 25](#), and that only the first and final nodes are so terminated.

Note: Some equipment has built-in pull up and pull down resistors which in some cases can be switched in or out of circuit. Such resistors must be removed or switched out of circuit for all but the instruments at each end of the line.

Intermittent Communications

1. Verify wiring paying particular attention to screening.
2. Ensure that the maximum line length of the transmission line has not been exceeded for the baud rate in use.
3. Ensure that the maximum line length of the transmission line has not been exceeded for the baud rate in use.
4. Ensure that the final node on the transmission line (no matter what type of instrument it is) is terminated correctly using three resistors as shown in [Figure 25](#), and that only the first and final nodes are so terminated.

DeviceNet

Introduction

DeviceNet has been designed as a low level network for communication between Programmable Logic Controllers (PLCs) and devices such as switches and IO devices. Each device and/or controller is a node on the network. EPower controller can be included in a DeviceNet installation using the DeviceNet interface module plugged into the communications slot. This module is an Unconnected Message Server (UCMM) capable device. The UCMM supports the Unconnected Explicit Message Request Port. EPower controller, in common with other Eurotherm controllers, has available a large number of potential parameters but practical systems are constrained by the total I/O space available in the master being used and by the amount of traffic permissible on the network. A limited number of pre defined parameters have, therefore, been made available in EPower controller but it is possible to add non defined parameters as required by a particular process. This is described in "Configuring the Data Exchange" on page 72.

Specific hardware must be used for the master – examples used in this chapter are Allen Bradley SLC500/03 processor with 1747-SDN scanner module and 1770-KFD RS232 interface with Rockwell RSLinx, RSNetWorx and RSLogic500.

It is not within the scope of this manual to describe the DeviceNet standard and for this you should refer to the DeviceNet specification which may be found at www.odva.org. In practice it is envisaged that EPower controller units will be added to an existing DeviceNet network. This chapter, therefore, is designed to provide practical help to set up EPower controller units on a DeviceNet network using one of the masters listed above.

There are 5 stages to setting up a network:

Physical Wiring	"DeviceNet Wiring" on page 70
Setting up EPower controller units	"Setting Up the EPower Controller Unit" on page 71
Setting up the master using EDS files	"Setting up the Master" on page 75
Configuring the data exchange	"Configuring the Data Exchange" on page 72
Establishing communications	"Establishing Communications" on page 75

EPower Controller DeviceNet Features

The DeviceNet implementation features in EPower controller include:

- Software selectable 125K, 250K, 500K baud rates
- Software selectable node address
- Optically isolated CAN interface
- 5-position open style connector
- Field pluggable option
- Group 2 only device

- Polled & Explicit I/O messaging connection
- Static I/O Assembly object

DeviceNet Wiring

A DeviceNet installation will consist of a Trunk Line installed around a process. This trunk line should be installed, including the correct termination resistors, in accordance with the DeviceNet specification. Devices, including EPower controller units, may be connected to this trunk line via Drop Lines. Each connection is referred to as a Node. Power to all devices will be supplied on the trunk line again in accordance with the DeviceNet specification.

Drop lines up to 6m (20 feet) each are permitted, allowing up to 64 nodes to be attached. DeviceNet allows branching structures only on a drop line. Termination resistors* should never be installed at the end of a drop line, only at the ends of the trunk line.

Figure 31 shows an example of two EPower controller units connected to an existing DeviceNet trunk line. All devices are similarly connected to the network.

According to the DeviceNet standard two types of cable may be used. These are known as Thick Trunk and Thin Trunk. For long trunk lines it is normal to use Thick trunk cable. For drop lines thin trunk cable is generally more convenient being easier to install. The table below shows the relationship between cable type, length and baud rate.

Network length	Varies with speed. Up to 400m possible with repeaters		
Baud Rate Mb/s	125	250	500
Thick trunk	500m (1,640ft)	250m (820ft)	100m (328ft)
Thin trunk	100m (328ft)	100m (328ft)	100m (328ft)

DeviceNet Trunk Cable

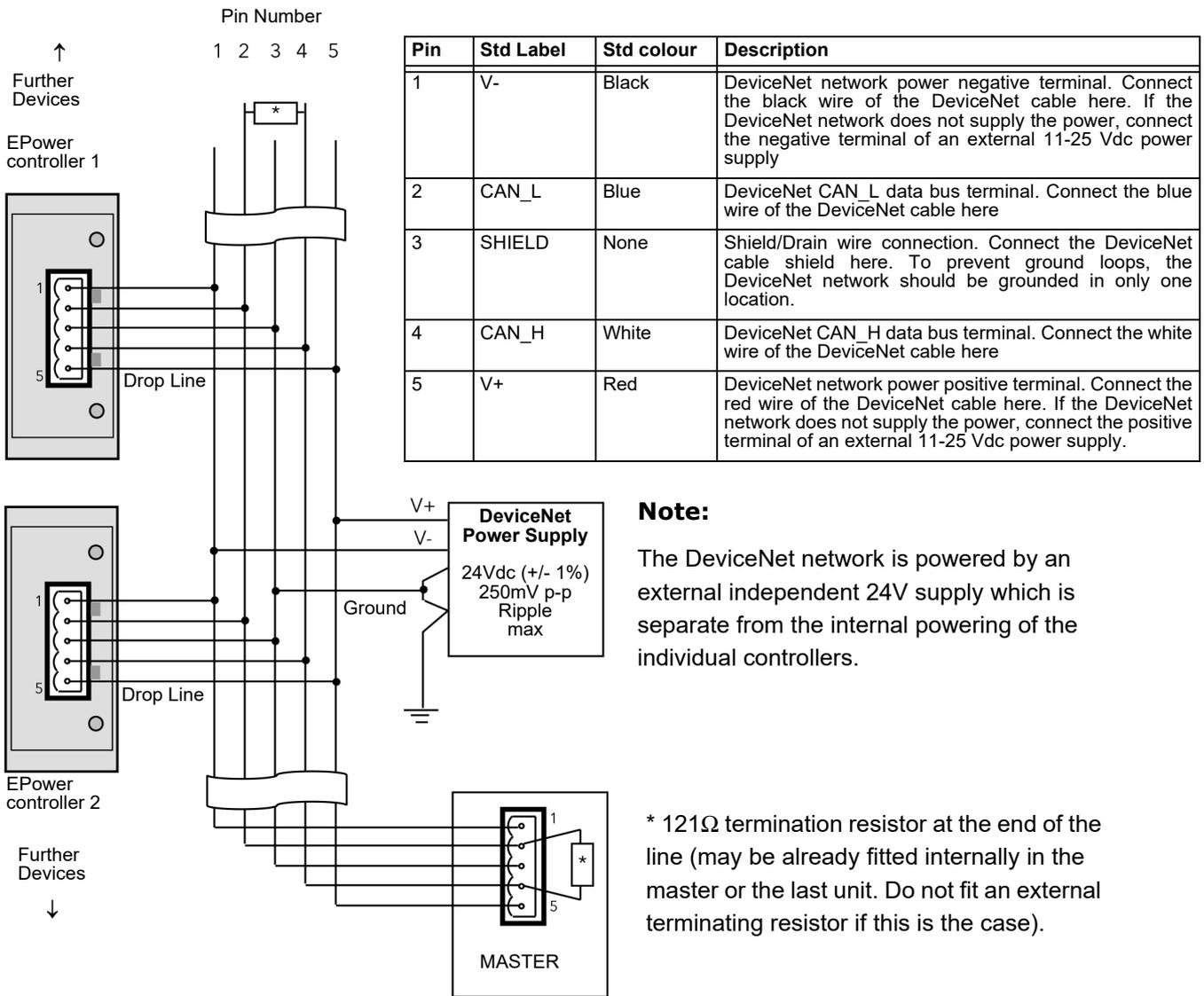


Figure 31 Example of DeviceNet Installation

Setting Up the EPower Controller Unit

The configuration for DeviceNet is slightly different for each type of controller but, having selected DeviceNet, there are only two parameters to set up – Baud rate and Address.

Valid Baud rates are 125k, 250k and 500k, and addresses may be from 0 to 63. Generally use 500k unless the network is longer than 100m. There is no priority in the addressing – all addresses are treated equally.

Unit Address

The unit address may be set in iTools or the EPower controller user interface. The parameter is called 'Address' which may be found in the 'Comms' list and may be changed in Engineer level. This is further described in the EPower controller User Guide (HA179769). The unit is shipped with a default address of 1. This is within the address range of the DeviceNet protocol (0 to 63), so if the unit is inadvertently inserted into the network without a new address having been set, the bus may be affected.

Note: After changing the DeviceNet address, the EPower controller unit should be powered off and on again, to allow correct initialisation to take place.

To set the address using iTools, open the Comms list and double click the 'User' sub-folder to open the list of parameters. Enter the value for the Comms Address.

Baud Rate

This can also be set up in iTools or through the EPower controller user interface. The parameter is called 'Baud' and is found in the 'Comms' list and can only be changed in Configuration level. This is further described in the EPower Controller User Guide (HA179769).

Data Exchange Mapping

Up to 32 input and 16 output variables may be included in the DeviceNet data exchange. By default, the most frequently used values are included, but it is possible to select other parameters within the unit. The default mapping is as follows:

Input Parameter	Byte Offset	Output Parameter	Byte Offset
Main PV (Network 1)	0	Main Setpoint (Network 1)	0
Main PV (Network 2)	2	Main Setpoint (Network 2)	2
Main PV (Network 3)	4	Main Setpoint (Network 3)	4
Main PV (Network 4)	6	Main Setpoint (Network 4)	6

The total length of both the default input and output data assemblies is, therefore, 8 bytes each.

To set up the EPower controller unit so that the desired parameters can be read and written involves setting up the INPUT and OUTPUT data tables. This is carried out using iTools.

Configuring the Data Exchange

The DeviceNet master may be required to work with many diverse slaves from different manufacturers and with different functions. Furthermore EPower controller units contain many parameters most of which will not be required by the network master for a particular application. It is, therefore, necessary for the user to define which Input and Output parameters are to be available on the DeviceNet link. The master may then map the selected device parameters into the PLC input/output registers, or, in the case of a supervisory (SCADA) package, to a personal computer.

Values from each slave, 'Input Data', are read by the master, which then runs a control program. The master then generates a set of values, 'Output Data', into a pre-defined set of registers to be transmitted to the slaves. This process is called an 'I/O data exchange' and is repeated continuously, to give a cyclical I/O data exchange.

The Input/Output definitions for DeviceNet are configured using iTools in the same way as for Profibus. Select the 'Fieldbus I/O Gateway' tool from the lower toolbar, and an editor screen will appear similar to the picture below.

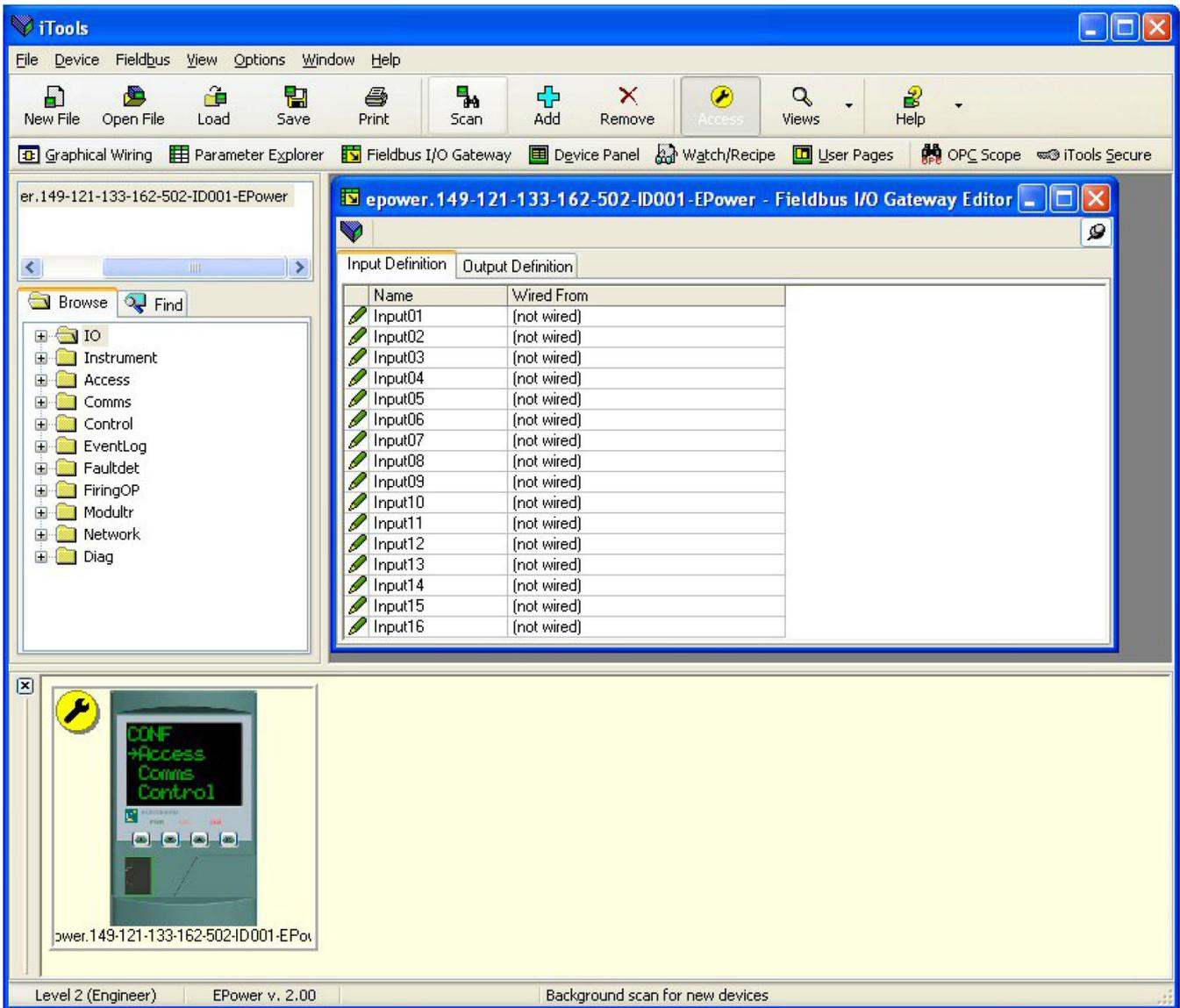


Figure 32 The I/O (Fieldbus I/O Gateway) Editor in iTools

There are two tabs within the editor, one for the definition of Inputs, and the other for Outputs. 'Inputs' are values sent from the EPower controller to the DeviceNet master, for example, alarm status information or measured values, i.e. they are readable values. 'Outputs' are values received from the master and used by the EPower controller, for example setpoints written from the master to EPower controller. Note that Outputs are written on every DeviceNet cycle, which is frequent, of the order of 10-100mS, and so values from DeviceNet will overwrite any changes made on the EPower controller keypad unless special measures are taken to prevent this.

Configuring Data Exchange continued:

The procedure for selecting variables is the same for both input and output tabs. Double click the next available position in the input or output data and select the variable to assign to it. A pop-up provides a browser from which a list of parameters can be opened. Double click the parameter to assign it to the input definition. Note that you should assign inputs and outputs contiguously, as a 'not wired' entry will terminate the list even if there are assignments following it. Figure 33 shows an example of the pop-up and the input list produced.

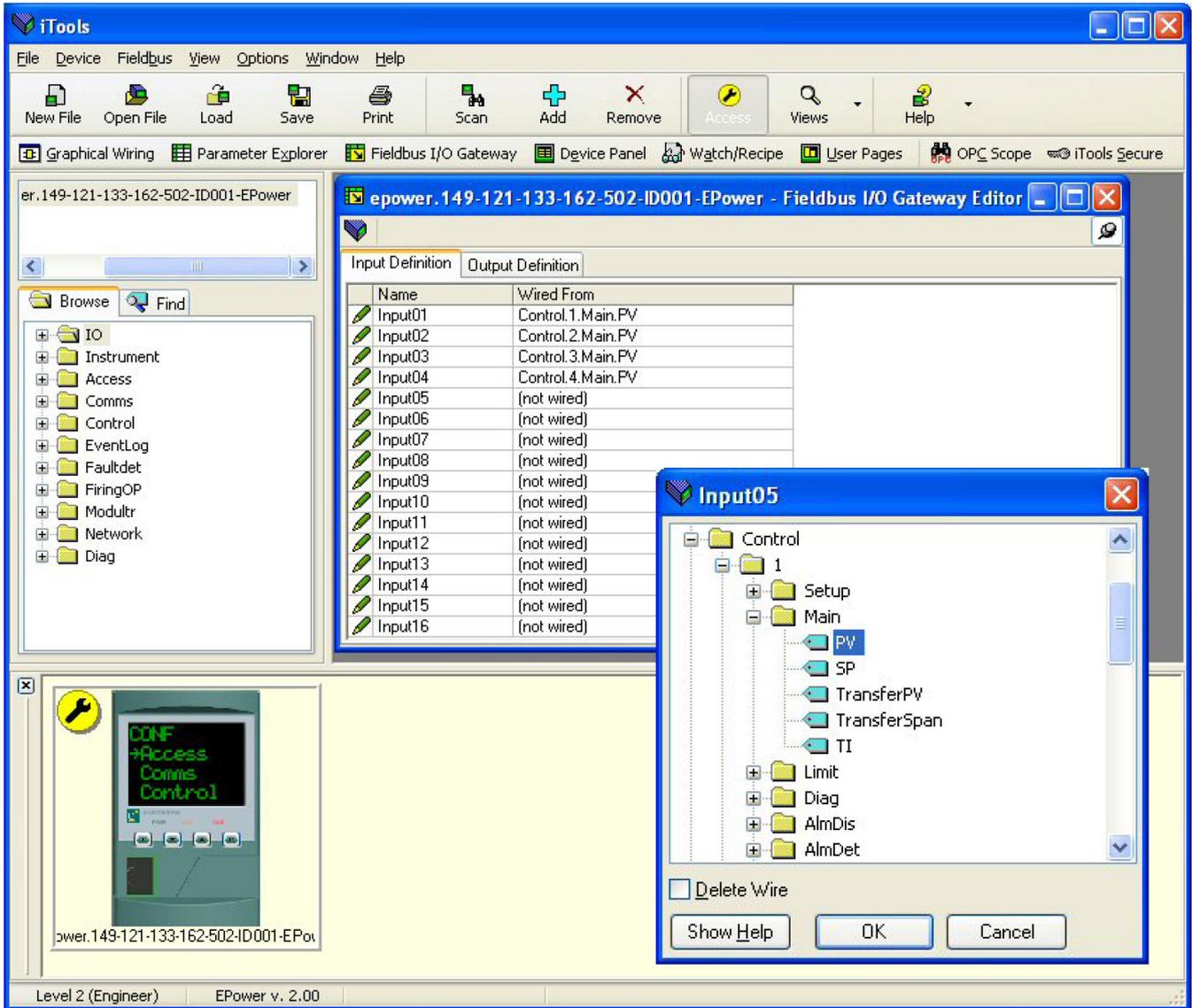


Figure 33 Selecting an Input Value and Example of an Input List

When the list is populated with the variables you require, note how many 'wired' entries are included in the input and output areas as this will be needed when setting up the DeviceNet Master. In the example above, there are four input values, each of two bytes in length, so a total of 8 bytes of data. Note this number, as it is required when setting the I/O length when configuring the DeviceNet master.

Note: No checks are made that output variables are writeable, and if a read only variable is included in the output list any values sent to it over DeviceNet will be ignored with no error indication.

Once the changes have been made to the I/O lists, they must be downloaded to the EPower controller unit.

This is done with the button on the top left of the I/O Editor marked . The EPower controller Unit will need to be powered off and on again once this has been done for the changes to register.

The next step in the process is to configure the DeviceNet master.

Setting up the Master

An example of a master may be a 1747-SDN scanner module from Allen Bradley. In this case use RSLinx and the Tools/Node Commissioning on RSNetWorx to set up the Scanner address and Baud Rate at which the network is to run. Baud rate cannot be changed 'on-line' it is only changed by closing down and re-starting the network.

Register all the required Eurotherm Electronic Data Sheets using the EDS Wizard in the Tools menu of RSNetWorx.

EDS Files are available from Eurotherm as EPOWER.EDS from www.eurotherm.com.

Note: the EDS file is unique and applies to the specific device. The device itself, not the .EDS file, is configured for each DeviceNet application.

Establishing Communications

With the DeviceNet network correctly wired up and powered, and the scanner and controllers configured with valid unique addresses and the same baud rate, communications will commence. If there is no communications check the common baud rate, unique addresses, 24V supply, the wiring, the termination resistors and finally the devices themselves.

If the Input/Output definitions have been changed from the default settings, it will be necessary to enter the lengths of the input and output data areas noted during their configuration as part of the master setup procedure.

At this stage communications is active and will be displayed by the LED indicators on the DeviceNet communications module. At this stage though it is only 'Hardware' communications and there is no transfer of data.

Data transfer has to be set up as a separate operation which involves both setting up EPower controller so that it knows what parameters it has to handle and setting up the scanner to make use of these parameters.

Parameters are either INPUT parameters read by the Scanner or OUTPUT parameters written by the scanner.

Data Formats

Data is returned as 'scaled integers', such that 999.9 is returned or sent as 9999; 12.34 is encoded as 1234. The control program in the DeviceNet master must convert the numbers into floating point values if required.

Explicit Messaging

It is possible to access any parameter in the EPower controller unit by means of 'explicit messaging', regardless of whether it has been included in the DeviceNet input/output data assembly. To do this, it is necessary to configure an explicit connection in the DeviceNet master. Then, to access parameters, use the ADI object (Class 0A2 hex), using an instance number equal to the Modbus address of the parameter. A list of Modbus addresses is given in the EPower Controller User Guide (HA179769). The 'Get Attribute Single' (OEhex) and 'Set Attribute Single' (010hex) services are used to retrieve and set values, in each case being applied to attribute 5 of the ADI object.

The EDS File

The EDS (Electronic Data Sheet) file for EPower controller is named EPOWER.EDS and is available from your supplier, or on the support disc supplied with the product, packaged with the firmware upgrade tool. The EDS file is designed to automate the DeviceNet network configuration process by precisely defining vendor-specific and required device parameter information. Software configuration tools utilise the EDS files to configure a DeviceNet network.

Trouble-Shooting

No Communications:

- Check the wiring carefully, paying particular attention to the continuity of the CAN-H and CAN-L connections to the scanner. Ensure that the correct terminals have been wired to.
- Make sure that there is 11-25Vdc supplied to the V+ and V- terminals. The controller will not communicate without power supplied.
- Check the 'Comms' list in configuration level and, under 'User' check that the parameter 'Ident' is showing Devicenet. If not, the unit may not be fitted with the correct DeviceNet communications module or it is not recognised by the EPower controller unit.
- Check that all devices including the scanner card or module have the same Baud Rate.
- Check that the Node 'Address' in the 'Comms' list is correct and unique for the network configuration in use.
- Ensure that the network is correctly configured and the configuration has been downloaded correctly to the DeviceNet scanner.
- Verify the ESD file in use is correct by loading it into the DeviceNet Configuration tool. This will check the format.
- Verify that the maximum line length for the baud rate in use is not exceeded (see table in "DeviceNet Wiring" on page 70).
- Ensure that the both ends of the DeviceNet network trunk line are correctly terminated (see wiring diagram). Ensure that no drop line devices have termination fitted.
- If possible, replace a faulty device with a duplicate and retest.

Ethernet/IP™

Introduction

EtherNet/IP™ (Ethernet/Industrial Protocol) is a 'producer-consumer' communication system used to allow industrial devices to exchange time-critical data. Such devices range from simple I/O devices such as sensors/actuators, to complex control devices such as robots and PLCs. The producer-consumer model allows the exchange of information between a single sending device (producer) and a large number of receiving devices (consumers) without having to send data multiple times to multiple destinations.

EtherNet/IP makes use of the CIP (Control & Information Protocol), common network, transport and application layers currently implemented by DeviceNet and ControlNet. Standard Ethernet and TCP/IP technology is used to transport CIP communications packets. The result is a common, open application layer on top of Ethernet and TCP/IP protocols.

The EPower controller can be included in an EtherNet/IP installation using the EtherNet/IP interface module plugged into the communications slot, for single port module (see "EtherNet/IP (Single Port)" on page 19) or dual port module (see "EtherNet/IP (Dual Port)" on page 20).

EPower controller, in common with other Eurotherm controllers, has available a large number of potential parameters but practical systems are constrained by the total I/O space available in the master being used and by the amount of traffic permissible on the network. A limited number of pre defined parameters have, therefore, been made available in EPower controller but it is possible to add non defined parameters as required by a particular process. This is described in "Data Exchange Mapping" on page 80.

Specific hardware must be used for the master such as an Allen-Bradley PLC.

It is not within the scope of this manual to describe the EtherNet/IP network and for this you should refer to information which may be found at www.odva.org - Navigate:- ODVA Technologies: EtherNet/IP: EtherNet/IP Library ("EtherNet/IP Infrastructure Guidelines" as well as other useful documents which may be found here).

There are 5 stages to setting up a network:

- Physical Wiring, "Ethernet/IP Wiring" on page 78
- Setting up EPower controller units, "Setting Up the EPower Controller Unit" on page 78
- Data exchange mapping, "Data Exchange Mapping" on page 80
- Setting up the master, "Setting Up the Master" on page 83 and "Acyclic (Explicit) Messaging" on page 84
- Establishing communications, "Establishing Communications" on page 84

EPower Controller EtherNet/IP Features

The EtherNet/IP implementation features in EPower controller include:

- 10/100Mbit, full / half duplex operation: auto sensing

- Galvanically isolated bus electronics
- Field pluggable option
- Polled & Explicit I/O messaging connection

Ethernet/IP Wiring

EtherNet/IP capability is provided by an interface board installed within the unit and provides a single RJ45 socket ("EtherNet/IP (Single Port)" on page 19) or dual RJ45 port sockets ("EtherNet/IP (Dual Port)" on page 20).

The EtherNet/IP port is a 10/100 Mbit, full or half duplex operation port and should be connected via an industrial switch with Cat5e (straight through) cable to a master device (e.g. PLC) via the standard RJ45 connector (max length 100m).

The interconnecting cables should be fitted with plugs provided with an outer metallic shell with the shell connected to the wire screen of the cable. See also "Wiring 3-Wire EIA485" on page 22 for suitable cables. This type of cable must be used to maintain EMC compliance.

Dual port socket supports both linear and ring network topology i.e. Device Level Ring (DLR).

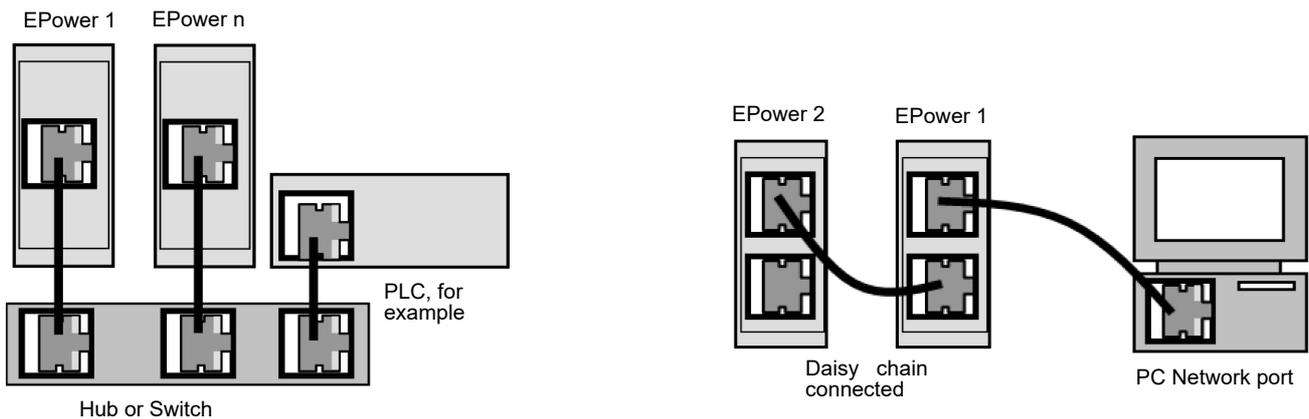


Figure 34 Ethernet/IP Wiring - Multiple EPower controller Units (Single port left, Double port right)

Setting Up the EPower Controller Unit

It is recommended that the communications settings for each instrument are set up before connecting it to any EtherNet/IP network. This is not essential but network conflicts may occur if the default settings interfere with equipment already on the network.

For the EtherNet/IP instrument the IP address, subnet mask, default gateway and DHCP enable need to be configured. These parameters are available in EPower controller units under different levels of access as detailed in the EPower Controller User Guide (HA179769).

Changing any one of these parameters may immediately move the instrument to a new network address. For this reason, it is recommended that such changes are made offline.

IP Addresses are usually presented in the form "abc.def.ghi.jkl". In the instrument Comms folder each element of the IP Address is shown and configured separately such that IPAdd1 = abc, IPAddr2 = def, IPAddr3 = ghi and IPAdr4 = jkl. This also applies to the SubNet Mask and Default Gateway IP Address.

Each Ethernet module contains a unique MAC address, normally presented as a 12 digit hexadecimal number in the format "aa-bb-cc-dd-ee-ff".

In EPower controller units MAC addresses are shown as 6 separate hexadecimal values on an EPower instrument itself or decimal values in iTools. MAC1 shows the first address value (aa), MAC2 shows the second address value (bb) and so on.

Dynamic Host Configuration Protocol (DHCP) Settings

This is set in configuration level by the DHCP Enable parameter.

IP addresses may be 'fixed' – set by the user, or dynamically allocated by a DHCP server on the network.

If IP Addresses are to be dynamically allocated the server uses the instrument MAC address to uniquely identify them.

Fixed IP Addressing

In the "Comms" folder of the instrument set the "DHCP enable" parameter to "Fixed". Set the IP address and SubNet Mask as required. This may be done in Engineer level.

Dynamic IP Addressing

In the "Comms" folder of the instrument set the "DHCP enable" parameter to "Dynamic". Once connected to the network and powered, the instrument will acquire its "IP address", "SubNet Mask" and "Default Gateway" from the DHCP Server and display this information within a few seconds.

Note that if the DHCP server does not respond (in common with other Ethernet appliances in this situation) the unit will not be accessible via the network.

Default Gateway

The "Comms" folder also includes configuration settings for "Default Gateway". These parameters will be set automatically when Dynamic IP Addressing is used. When fixed IP addressing is used these settings are only required if the instrument needs to communicate wider than the local area network.

[Figure 35](#) below shows the appearance of EtherNet/IP User Comms configuration parameters in iTools:

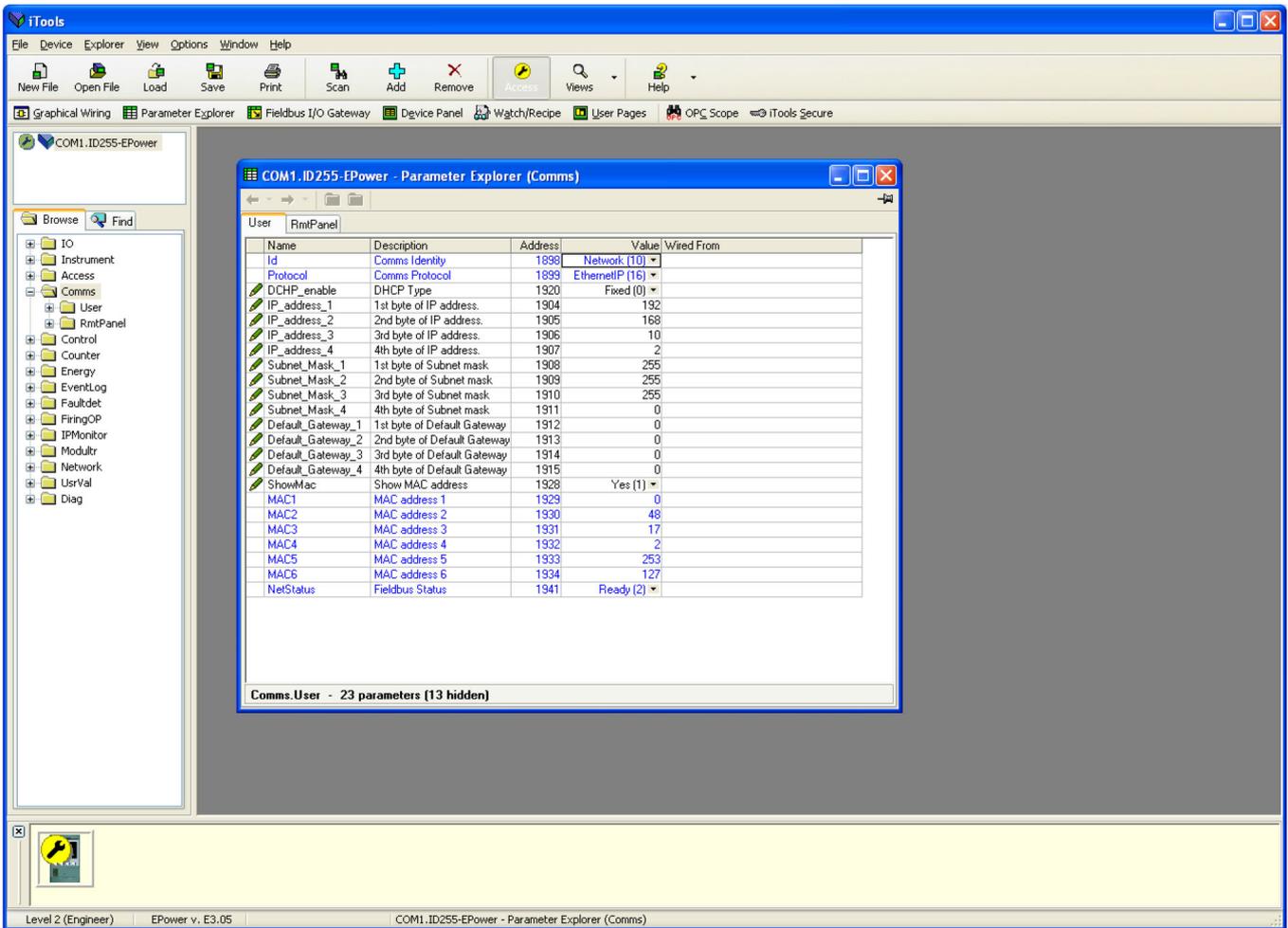


Figure 35 EtherNet I/P Comms Parameters

Data Exchange Mapping

Up to 32 input and 16 output parameter variables may be included in the EtherNet/IP cyclic (implicit) data exchange.

By default, the most frequently used values are included, but it is possible to select other parameters within the unit. The default mapping is as follows:

Input Parameter	Output Parameter
Main PV (Network 1)	Main Setpoint (Network 1)
Main PV (Network 2)	Main Setpoint (Network 2)
Main PV (Network 3)	Main Setpoint (Network 3)
Main PV (Network 4)	Main Setpoint (Network 4)

Input and Output Parameters are 16 bits (2 bytes) each.

To set up the EPower controller unit so that the desired parameters can be read and written involves setting up the INPUT and OUTPUT data tables. This is carried out using iTools.

Configuring the Cyclic (Implicit) Data Exchange

The EtherNet/IP master may be required to work with many diverse slaves from different manufacturers and with different functions. Furthermore EPower controller units contain many parameters most of which will not be required by the network master for a particular application. It is, therefore, necessary for the user to define which Input and Output parameters are to be available on the EtherNet/IP network. The master may then map the selected device parameters into the PLC input/output registers.

Values from each slave, 'Input Data', are read by the master, which then runs a control program. The master then generates a set of values, 'Output Data', into a pre-defined set of registers to be transmitted to the slaves. This process is called an 'I/O data exchange' and is repeated continuously, to give a cyclical I/O data exchange.

The Input/Output definitions for EtherNet/IP are configured using iTools in the same way as for DeviceNet or Profibus.

Select the 'Fieldbus I/O Gateway' tool from the lower toolbar, and an editor screen will appear similar to that shown in [Figure 36](#):

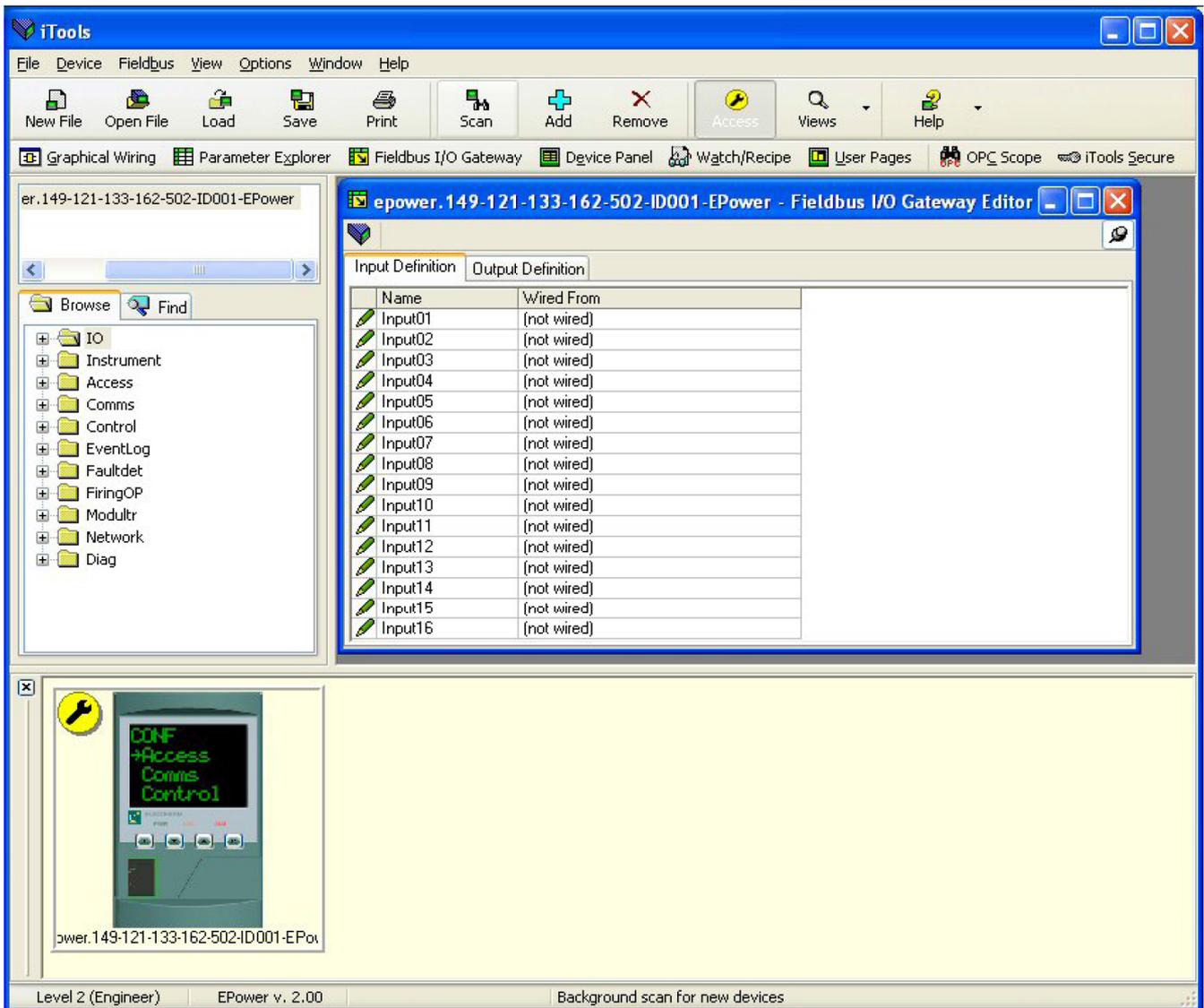


Figure 36 The I/O (Fieldbus I/O Gateway) Editor in iTools

There are two tabs within the editor, one for the definition of Inputs, and the other for Outputs. 'Inputs' are values sent from the EPower controller to the EtherNet/IP master, for example, alarm status information or measured values, i.e. they are readable values. 'Outputs' are values received from the master and used by the EPower controller, for example, setpoints written from the master to EPower controller. Note that Outputs are written on every EtherNet/IP cycle, which is frequent, of the order of 10-100mS, and so values from EtherNet/IP will overwrite any changes made on the EPower controller keypad unless special measures are taken to prevent this.

The procedure for selecting variables is the same for both input and output tabs. Double click the next available position in the input or output data and select the variable to assign to it. A pop-up provides a browser from which a list of parameters can be opened. Double click the parameter to assign it to the input definition. Note that you should assign inputs and outputs contiguously, as a 'not wired' entry will terminate the list even if there are assignments following it.

Figure 37 shows an example of the pop-up and the input list produced.

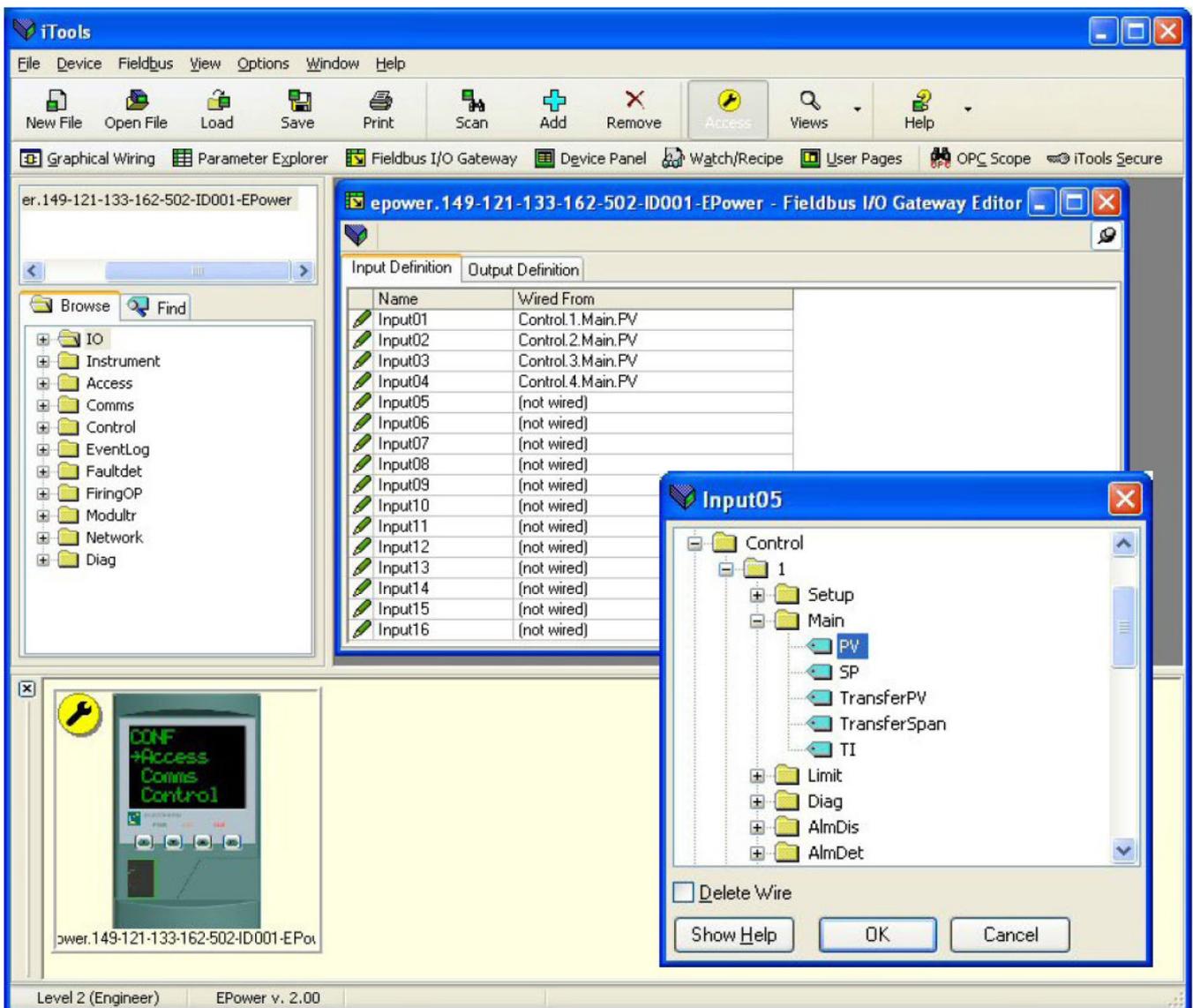


Figure 37 Selecting an Input Value and Example of an Input List

When the list is populated with the variables you require, note how many 'wired' entries are included in the input and output areas as this will be needed when setting up the EtherNet/IP Master. In the example above, there are four input values, each of two bytes in length, so a total of 8 bytes of data. Note this number, as it is required when setting the I/O length when configuring the EtherNet/IP master.

Note that no checks are made that output variables are writeable, and if a read only variable is included in the output list, any values sent to it over EtherNet/IP will be ignored with no error indication.

Once the changes have been made to the I/O lists, they must be downloaded to the EPower controller unit.

This is done with the button on the top left of the I/O Editor marked  .

The EPower controller Unit will need to be powered off and on again once this has been done for the changes to register.

The next step in the process is to configure the EtherNet/IP master.

Setting Up the Master

An example of a master may be a CompactLogix L23E QB1B PLC from Allen Bradley.

In this case RSLogix 5000 may be use to set up the PLC EtherNet/IP Master.

Cyclic (Implicit) Data Exchange

Using RSLogix 5000 as an example:

In I/O configuration, select "New Module" and select "Generic Ethernet module"

In the next dialogue window, RSLogix 5000 will ask for information regarding the communication to the EPower EtherNet/IP Slave module.

First enter a name for the EPower EtherNet/IP Slave module: eg 'EPower'.

This name will create a tag in RSLogix 5000, which can be used to access the memory location in the PLCs memory where the data for the EPower Slave module will be stored.

Next step is to select the "Comm Format", which tells RSLogix5000 the format of the data. Select Data-INT, which will represent the data as 16-bit values. (EPower I/O parameters, defined by the iTools Fieldbus I/O Gateway Editor, are 16 bit values).

I/O data is accessed in Input Instance 100 and Output Instance 150, so these values have to be entered as the instance values for input and output.

The size of the input connection and the output connection shall correspond to the size that has been defined by the 'iTools Fieldbus I/O Gateway Editor' Input and Output Definitions for the EPower slave.

That is:- Input size (in 16 bit values in this case) = Number of 'I/O Gateway' Input Parameter definitions.

Output size (in 16 bit values in this case) = Number of 'I/O Gateway' Output Parameter definitions.

The EPower EtherNet/IP Slave module does not have a configuration assembly instance, but RSLogix5000 requires a value for this anyway. An instance value of 0 is not a valid instance number, but any non-zero value will work, so use a value 199. The data size of the configuration instance has to be set to 0, otherwise the configuration instance will be accessed and the connection will be refused.

As a final step enter the IP address that has been configured for the EPower EtherNet/IP slave module.

Summary: Cyclic (implicit) I/O Data Exchange setup information:

	Assembly Instance	Data Size
INPUT	100	2 Bytes per “iTools Fieldbus I/O Gateway” Input Parameter Definition
OUTPUT	150	2 Bytes per “iTools Fieldbus I/O Gateway” Output Parameter Definition
CONFIGURATION	199 *	0

Note: EPower EtherNet/IP does not have a configuration assembly instance: So use 199 (assembly instance has to be non zero) and set data size to 0.

Acyclic (Explicit) Messaging

Acyclic (or explicit) messaging is used to transfer data that does not require continuous updates.

It is possible to access any parameter in the EPower controller unit by means of ‘explicit messaging’, regardless of whether it has been included in the EtherNet/IP input/output data assembly. To do this, it is necessary to configure an explicit connection in the EtherNet/IP master.

To access parameters, use the ADI object (Class 0xA2 hex), using an instance number equal to the Modbus address of the parameter.

A list of Modbus addresses is given in the EPower Controller User Guide (HA179769).

The ‘Get Attribute Single’ (0Ehex) and ‘Set Attribute Single’ (010hex) services are used to retrieve (read) and set (write) values, in each case being applied to attribute 5 (‘Value’) of the ADI object.

Summary: Acyclic (explicit) I/O Data Exchange setup information:

Message type: CIP Generic

Service Type: [Service Code]: Get Attribute Single (read): [0x0E]

Set Attribute Single (write): [0x10]

Class:	ADI Object:	[0xA2]
Instance:	Parameter Modbus Address	
Attribute:	Value:	[0x05]

Establishing Communications

Communications will commence when the EtherNet/IP network is correctly cabled and powered, the Master (eg PLC) and Slave (EPower) EtherNet/IP modules are configured with valid unique IP addresses and I/O parameter data definitions are setup.

The Input/Output definitions need to be matched with Master (eg PLC) data registers.

At this stage communications is active and will be displayed by the LED indicators on the EPower EtherNet/IP communications module.

Parameters are either INPUT parameters read by the EtherNet/IP Master or OUTPUT parameters written by the EtherNet/IP Master.

Data Formats

Data is returned as 'scaled integers', such that 999.9 is returned or sent as 9999; 12.34 is encoded as 1234. The control program in the EtherNet/IP master must convert the numbers into floating point values if required.

The EDS File

The EtherNet/IP EDS (Electronic Data Sheet) file for EPower controller is named:

- 368-0164-EDS_ABCC_EIP_V_2_1.eds, for single port version ("EtherNet/IP (Single Port)" on page 19)
- 368-0164-EDS_ABCC_EIP2P_V_1_5.eds, for dual port version ("EtherNet/IP (Dual Port)" on page 20)

Which is available from your supplier, or electronically by going to Web site (www.eurotherm.com).

The EDS file is designed to automate the EtherNet/IP network configuration process by precisely defining the required device parameter information. Software configuration tools utilise the EDS file to configure an EtherNet/IP network.

Troubleshooting

No Communications:

- Check the cabling carefully, ensure that Ethernet plugs are fully located in the sockets.
- Check the 'Comms' list in configuration level and, under 'User' check that the parameter 'Ident' is showing Network and the 'Protocol' is showing EthernetIP. If not, the unit may not be fitted with the correct EtherNet/IP communications module or it is not recognised by the EPower controller unit.
- Check that the 'IP Address', 'Subnet Mask' and 'Gateway' in the 'Comms' list are correct and unique for the network configuration in use.
- Ensure that the network is correctly configured and the configuration has been downloaded correctly to the EtherNet/IP Master Module.
- Ensure that the EtherNet/IP Master Module Input and Output Parameter mapping is correctly matched. If the master is attempting to read (input) or write (output) more data than has been registered on the EPower slave, using the iTools I/O Gateway Editor, the EPower slave will refuse the connection.
- If possible, replace a faulty device with a duplicate and retest.

CC-Link

Introduction

CC-Link is an open fieldbus and control network. It provides for communication between Programmable Logic Controllers (PLCs) and devices such as switches and IO devices. Each device and/or controller is a station on the network.

EPower controller can be included in a CC-Link installation using the CC-Link interface module plugged into the communications slot, see "CC-Link" on page 20.

EPower controller, in common with other Eurotherm controllers, has available a large number of potential parameters but practical systems are constrained by the total I/O space available in the master being used and by the amount of traffic permissible on the network. A limited number of pre defined parameters have, therefore, been made available in EPower controllers but it is possible to add non defined parameters as required by a particular process. This is described in "Configuring the Data Exchange" on page 91.

Specific hardware must be used for the master – examples used in this chapter are Mitsubishi FX2N-16MR PLC with a FX2N-16CCL-M CC-Link Master Module and Q Series Mitsubishi PLC with a QJ61BT11N CC-Link Master Module.

It is not within the scope of this manual to describe the CC-Link network and for this you should refer to information which may be found at www.cc-link.org.

In practice it is envisaged that EPower controller units will be added to an existing CC-Link network. This chapter, therefore, is designed to provide practical help to set up EPower controller units on a CC-Link network using, as an example, one of the masters listed above.

There are 5 stages to setting up a network:

- Physical Wiring, "CC-Link Wiring" on page 88
- Setting up EPower controller units, "Setting Up the EPower Controller Unit" on page 89
- Setting up the data exchange, "Data Exchange Mapping" on page 90 and "Configuring the Data Exchange" on page 91
- Setting up the master using example PLC setup project files, "Setting Up the Master" on page 94
- Establishing communications, "Establishing Communications" on page 96

EPower Controller CC-Link Features

The CC-Link implementation (V1.1) features in EPower controller include:

- Software selectable 156K, 625K, 2.5M, 5M and 10M baud rates
- Software selectable Station Number (user address)
- Indication of number of Occupied Stations
- Optically isolated CC-Link interface

- Open style connector
- Field pluggable option
- Polled I/O data read/write connection

CC-Link Wiring

This section is intended to provide general information on wiring between stations.

For a full description refer to www.cc-link.org.

EPower currently supports V1.1 features of CC-Link.

Please also refer to "CC-Link" on page 20 for pinout connections on the EPower unit.

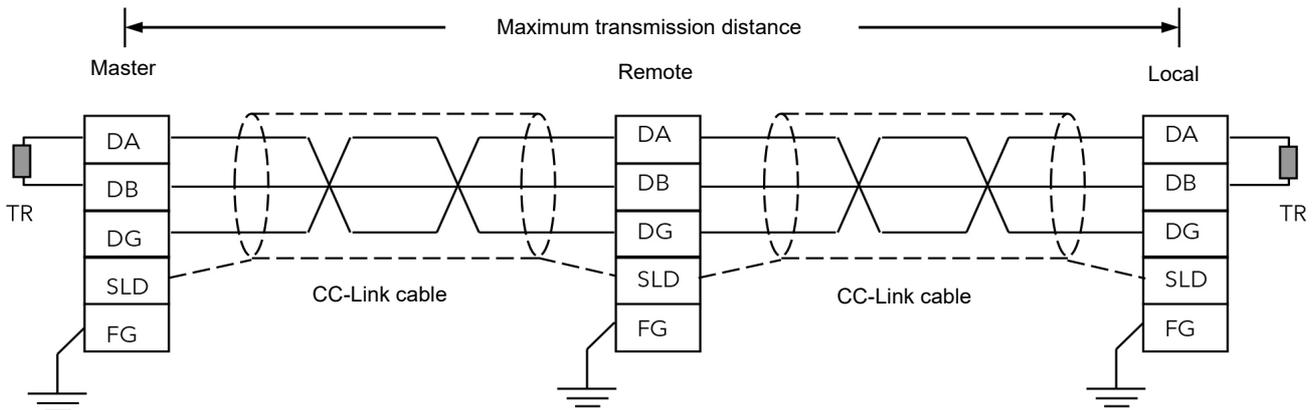


Figure 38 Single trunk line showing connection of terminating resistors

Connect a terminating resistor (TR) at each end of the line. For a single line with no repeaters the resistor values are shown in the table below:

Terminating resistor (TR)	Cable
110 ohm $\pm 5\%$ 1/2W	V 1.10 compatible CC-Link dedicated cable
	V 1.00 compatible CC-Link dedicated cable
130 ohm $\pm 5\%$ 1/2W	V 1.00 compatible CC-Link dedicated high performance cable

Note: If a repeater is used, use the terminating resistor in the repeater module. See www.cc-link.org for further details.

Protective Earth

For best results ground the FG terminals independently to the protective ground conductor (ground resistance 100Ω or less) as shown in [Figure 38](#).

Cable Shield

Connect both ends of the cable shield to SLD of each module as shown in [Figure 38](#).

Induced Noise

Keep signal line as far away as possible from power lines and high voltage devices.

Maximum Transmission Distance

Maximum Transmission Distance means the total cable length from end to end with multi-dropped connection. The maximum distance depends upon the communication speed and the type of CC-Link dedicated cable as shown in the table below:

Communication speed (Baud Rate)	Maximum transmission distance	
	V1.10 compatible CC-Link dedicated cable. V1.00 compatible CC-Link dedicated high performance cable.	V1.00 compatible CC-Link dedicated cable.
10Mbps	100m	100m
5Mbps	160m	150m
2.5Mbps	400m	200m
625kbps	900m	600m
156kbps	1200m	1200m

Setting Up the EPower Controller Unit

There are only two CC-Link configuration parameters to set up – Baud rate and Address (Station Number).

Valid Baud rates are 156k, 625k, 2.5M, 5M and 10M, and addresses (station numbers) may be from 1 to 64.

An additional informational parameter is also presented: Occupied Stations: The value of Occupied Stations indicates how many network station numbers are occupied by this device.

Unit Address (CC-Link Station Number)

The unit address or Station Number for CC-Link, may be set in iTools or the EPower controller user interface.

The parameter is called 'Address' which may be found in the 'Comms' list and may be changed in Engineer level. This is further described in the EPower controller User Guide. The unit is shipped with a default address (station number) of 1. This is within the address range of the CC-Link protocol (1 to 64), so if the unit is inadvertently inserted into the network without a new address having been set, the bus may be affected.

Note: After changing the CC-Link address (Station Number), the EPower controller unit should be powered off and on again, to allow correct initialisation to take place.

To set the address (station number) using iTools, open the Comms list and double click the 'User' sub-folder to open the list of parameters.

Enter the value for the Comms Address.

Baud Rate

This can also be set up in iTools or through the EPower controller user interface. The parameter is called 'Baud' and is found in the 'Comms' list and can only be changed in Configuration level. This is further described in the EPower controller User Guide (HA179769).

Occupied Stations

This is an informational parameter. The value of Occupied Stations indicates how many network station numbers are occupied by this device.

The next available network station number ('Address') is this device's station number ('Address') plus the number of occupied stations.

For example:- If the station number of this device is 4 and it occupies 2 stations then the next available network device station number ('Address') would be 6.

The value of occupied stations is dependant upon the size of the mapped process data as shown in the following table:

Number of Occupied Stations	Number of Input Definitions (word (2 byte) parameters to be read by the master)	Number of Output Definitions (word (2 byte) parameters to be written by the master)
1	Upto 3	Upto 4
2	Upto 7	Upto 8
3	Upto 11	Upto 12
4	Upto 15	Upto 16

Where Number of Input and Output definitions are the number of input (read) and output (write) parameters setup using the iTools 'Fieldbus I/O Gateway' tool. (See Data Exchange Mapping section below).

Note: Setting up 16 Input Definitions will cause an error condition: The "User Status" parameter will indicate this by reporting ">15 Input" and the CC-Link Module ERR LED will be illuminated.

Data Exchange Mapping

On CC-Link, data is divided into two categories as follows:

Bit Area

Data is accessed on a bit by bit basis. Data is commonly referred to as RX #nn (Slave -> Master) and RY #nn (Master -> Slave) where 'nn' represents an addressable point (i.e. a single bit) in the Bit Area.

Note: On EPower the bit area is NOT utilised EXCEPT for the CC-Link System Area. CC-Link System Area Location and functionality is described in a later section.

Word Area

Data is accessed as 16-bit words. Data is commonly referred to as RWr #nn (Slave->Master) and RWw #nn (Master->Slave) where 'nn' represents an addressable point (i.e. a word) in the Word Area.

Up to 15 input and 16 output variables may be included in the CC-Link data exchange. These are mapped into the Word Area.

By default, the most frequently used values are included, but it is possible to select other parameters within the unit. The default mapping is as follows:

Input Parameter	Byte Offset (from start of Word Area)	Output Parameter	Byte Offset (from start of Word Area)
Main PV (Network 1)	0	Main Setpoint (Network 1)	0
Main PV (Network 2)	2	Main Setpoint (Network 2)	2
Main PV (Network 3)	4	Main Setpoint (Network 3)	4
Main PV (Network 4)	6	Main Setpoint (Network 4)	6

Input and Output Parameters are a word (2 bytes) each.

To set up the EPower controller unit so that the desired parameters can be read and written involves setting up the INPUT and OUTPUT data tables. This is carried out using iTools.

See “Occupied Stations” section above for relationship between the number of Input and Output parameters and Occupied Stations.

Configuring the Data Exchange

The CC-Link master may be required to work with many diverse slaves from different manufacturers and with different functions. Furthermore EPower controller units contain many parameters most of which will not be required by the network master for a particular application. It is, therefore, necessary for the user to define which Input and Output parameters are to be available on the CC-Link link. The master may then map the selected device parameters into the PLC input/output registers.

Values from each slave, ‘Input Data’, are read by the master, which then runs a control program. The master then generates a set of values, ‘Output Data’, into a pre-defined set of registers to be transmitted to the slaves. This process is called an ‘I/O data exchange’ and is repeated continuously, to give a cyclical I/O data exchange.

The Input/Output definitions for CC-Link are configured using iTools in the same way as for DeviceNet or Profibus. Select the ‘Fieldbus I/O Gateway’ tool from the lower toolbar, and an editor screen will appear similar to the picture below.

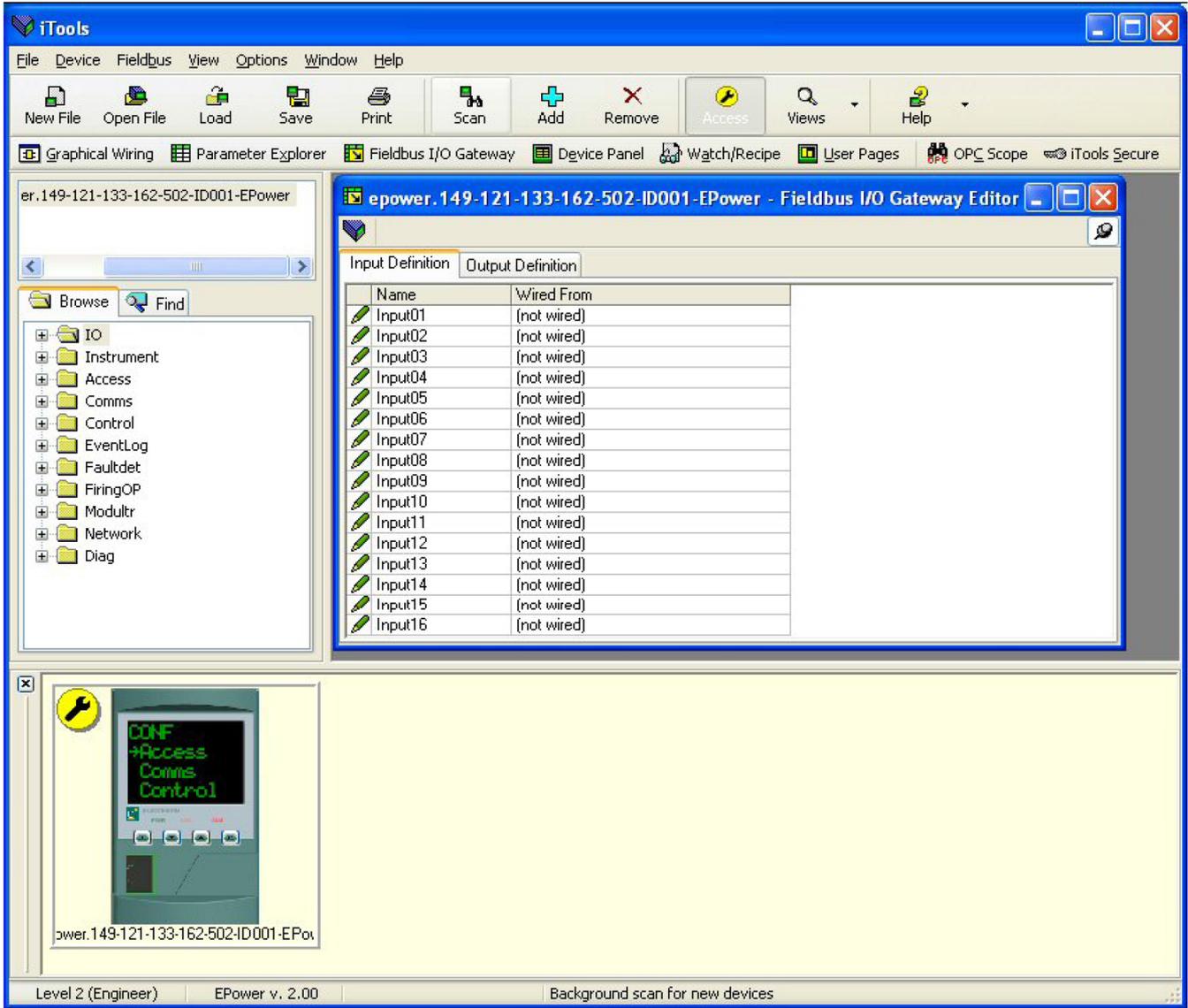


Figure 39 The I/O (Fieldbus I/O Gateway) Editor in iTools

There are two tabs within the editor, one for the definition of Inputs, and the other for Outputs. 'Inputs' are values sent from the EPower controller to the CC-Link master, for example, alarm status information or measured values, i.e. they are readable values. 'Outputs' are values received from the master and used by the EPower controller, for example setpoints written from the master to EPower controller. Note that Outputs are written on every CC-Link cycle, which is frequent, of the order of 10-100mS, and so values from CC-Link will overwrite any changes made on the EPower controller keypad unless special measures are taken to prevent this.

Note: Input 16 is not available for use by the CC-Link Master.

(The last word is reserved for use by the CC-Link Module fitted in the EPower controller. Setting up 16 Input Definitions will cause an error condition: The "User Status" parameter will indicate this by reporting ">15 Input" and the CC-Link Module ERR LED will be illuminated).

The procedure for selecting variables is the same for both input and output tabs. Double click the next available position in the input or output data and select the variable to assign to it. A pop-up provides a browser from which a list of parameters can be opened. Double click the parameter to assign it to the input definition. Note that you should assign inputs and outputs contiguously, as a 'not wired' entry will terminate the list even if there are assignments following it. Figure 40 shows an example of the pop-up and the input list produced.

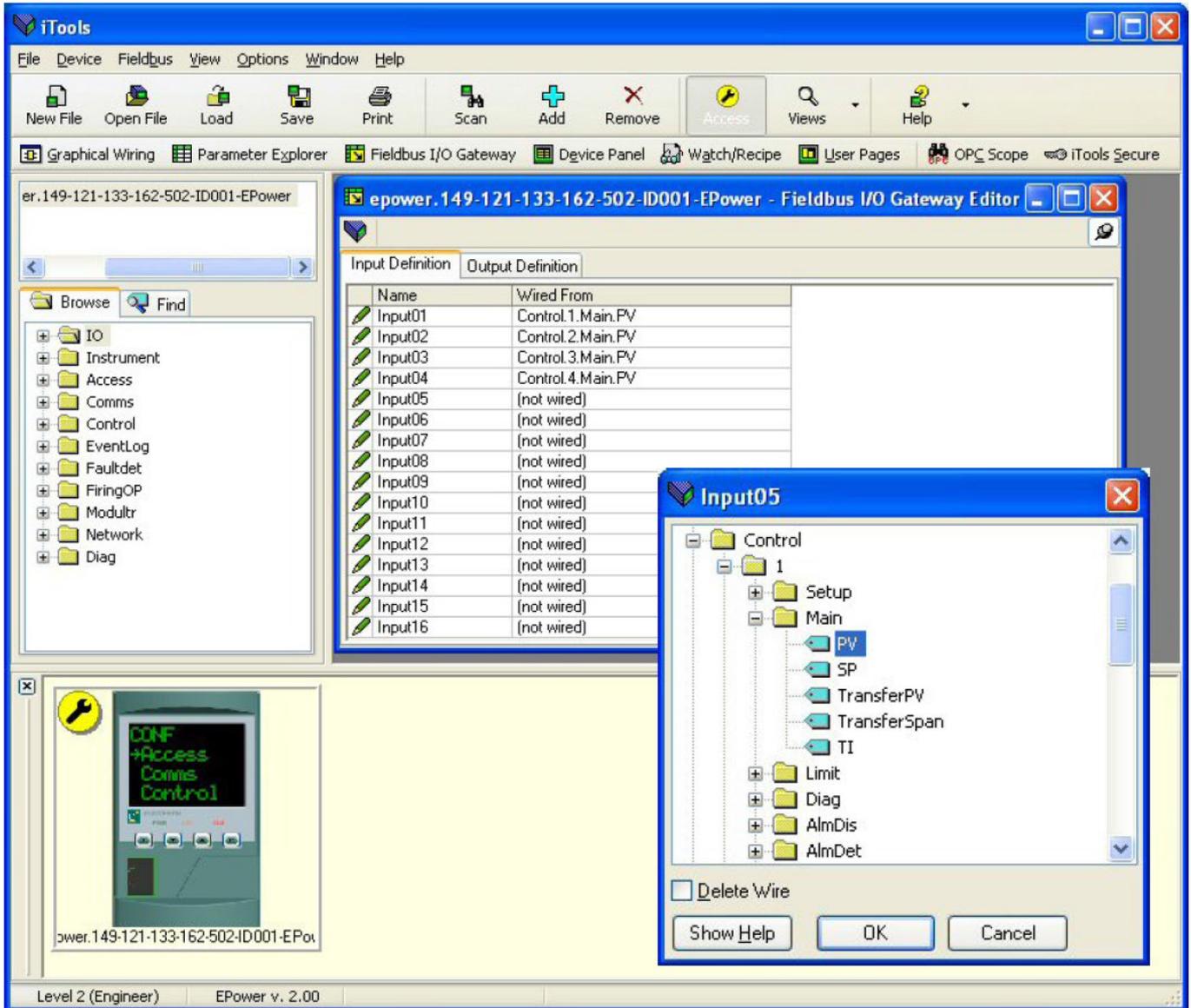


Figure 40 Selecting an Input Value and Example of an Input List

When the list is populated with the variables you require, note how many 'wired' entries are included in the input and output areas as this will be needed when setting up the CC-Link Master.

In the example above, there are four input values, each a word (2 bytes) in length, so a total of 4 words of data are to be read from the CC-Link Word Area.

Note that no checks are made that output variables are writeable, and if a read only variable is included in the output list any values sent to it over CC-Link will be ignored with no error indication.

Once the changes have been made to the I/O lists, they must be downloaded to the EPower controller unit.

This is done with the button on the top left of the I/O Editor marked .

The EPower controller Unit will need to be powered off and on again once this has been done for the changes to register.

The next step in the process is to configure the CC-Link master.

Setting Up the Master

Examples

An example of a master may be a Mitsubishi FX2N-16MR PLC with a FX2N-16CCL-M CC-Link Master module.

In this case Melsoft GX Developer FX may be used to setup the PLC and CC-Link master module.

Example GX Developer FX project files are available from Eurotherm as “EPower Example CC Link Setup 1 occupied station” and “EPower Example CC Link Setup 2 occupied stations” from www.eurotherm.com.

These example project files include PLC ladder programs that configure the PLC and CC-Link master module to read and write parameters from/to an EPower controller into/from PLC data registers.

“EPower Example CC Link Setup 1 occupied station” provides an example where 3 parameters are read and 3 are written. In this case the EPower Controller occupies 1 station.

“EPower Example CC Link Setup 2 occupied stations” provides an example where 6 parameters are read and 6 are written. In this case the EPower Controller occupies 2 stations.

An EPower Controller CC-Link slave appears as a “I/O Device” on the CC-Link network.

CC-Link System Area

An essential part of the CC-Link communication is the CC-Link System Area. This area holds various status flags.

System Area Layout

Slave -> Master		Master -> Slave	
Bit Offset	Contents	Bit Offset	Contents
0	(reserved)	0	(reserved)
1	(reserved)	1	(reserved)
2	(reserved)	2	(reserved)
3	(reserved)	3	(reserved)
4	(reserved)	4	(reserved)
5	(reserved)	5	(reserved)
6	(reserved)	6	(reserved)

Slave -> Master		Master -> Slave	
Bit Offset	Contents	Bit Offset	Contents
7	(reserved)	7	(reserved)
8	Initial Data Processing Request	8	Initial Data Processing Complete
9	Initial Data Setting Complete	9	Initial Data Setting Request
10	(reserved)	10	(reserved)
11	Remote READY	11	(reserved)
12	(reserved)	12	(reserved)
13	(reserved)	13	(reserved)
14	(reserved)	14	(reserved)
15	(reserved)	15	(reserved)

System Area Location

The System Area is located at the very end of the Bit Area as follows:

Point	Contents	Point	Contents
RX #0	CC – Link User Area (not utilised by EPower)	RY #0	CC – Link User Area (not utilised by EPower)
RX #1	CC – Link User Area (not utilised by EPower)	RY #1	CC – Link User Area (not utilised by EPower)
RX #3	CC – Link User Area (not utilised by EPower)	RY #3	CC – Link User Area (not utilised by EPower)
.....	CC – Link User Area (not utilised by EPower)	CC – Link User Area (not utilised by EPower)
.....	CC – Link User Area (not utilised by EPower)	CC – Link User Area (not utilised by EPower)
RX #Q – 18	CC – Link User Area (not utilised by EPower)	RY #Q – 18	CC – Link User Area (not utilised by EPower)
RX #Q – 17	CC – Link User Area (not utilised by EPower)	RY #Q – 17	CC – Link User Area (not utilised by EPower)
RX #Q – 16	CC-Link System Area (reserved)	RY #Q – 16	CC-Link System Area (reserved)
RX #Q – 15	CC-Link System Area (reserved)	RY #Q – 15	CC-Link System Area (reserved)
RX #Q – 14	CC-Link System Area (reserved)	RY #Q – 14	CC-Link System Area (reserved)
RX #Q – 13	CC-Link System Area (reserved)	RY #Q – 13	CC-Link System Area (reserved)
RX #Q – 12	CC-Link System Area (reserved)	RY #Q – 12	CC-Link System Area (reserved)
RX #Q – 11	CC-Link System Area (reserved)	RY #Q – 11	CC-Link System Area (reserved)
RX #Q – 10	CC-Link System Area (reserved)	RY #Q – 10	CC-Link System Area (reserved)
RX #Q – 9	CC-Link System Area (reserved)	RY #Q – 9	CC-Link System Area (reserved)
RX #Q – 8	CC-Link System Area: Initial Data Processing Request	RY #Q – 8	CC-Link System Area: Initial Data Processing Complete
RX #Q – 7	CC-Link System Area: Initial Data Setting Complete	RY #Q – 7	CC-Link System Area: Initial Data Setting Request
RX #Q – 6	CC-Link System Area (reserved)	RY #Q – 6	CC-Link System Area (reserved)
RX #Q – 5	CC-Link System Area: Remote READY	RY #Q – 5	CC-Link System Area (reserved)
RX #Q – 4	CC-Link System Area (reserved)	RY #Q – 4	CC-Link System Area (reserved)
RX #Q – 3	CC-Link System Area (reserved)	RY #Q – 3	CC-Link System Area (reserved)
RX #Q – 2	CC-Link System Area (reserved)	RY #Q – 2	CC-Link System Area (reserved)
RX #Q – 1	CC-Link System Area (reserved)	RY #Q – 1	CC-Link System Area (reserved)

Where #Q represents the number of addressable points in the Bit Area. Number of addressable points in the Bit Area is dependant upon the number of Occupied Stations as follows:

Occupied Stations	Number of Addressable points in Bit Area
1	32 bits
2	64 bits
3	96 bits
4	128 bits

For Example, if an EPower was setup to occupy 2 stations, the “Initial Data Processing Request” flag would be located at bit RX #56 (ie 64 – 8 = 56).

System Area Flag Handshaking

The CC-Link Master must undertake the following handshake procedure in order to place an EPower CC-Link Slave into its network status Active state.

On “Initial Data Processing Request” flag being Set (= 1)

Set (=> 1) “Initial Data Processing Complete” flag

Set (=> 1) “Initial Data Setting Request” flag

Otherwise

Clear (=> 0) “Initial Data Processing Complete” flag

On “Initial Data Setting Complete” flag being Set (= 1)

- Clear (=> 0) “Initial Data Setting Request” flag

The above handshake procedure is included in the example GX Developer FX project files (PLC ladder program) referred to earlier.

“Remote READY” flag: = 1: Normal Operation 0: Abnormal Operation

Establishing Communications

With the CC-Link network correctly wired up and powered, and the PLC and CC-Link modules configured with valid unique Station Numbers and the same baud rate, communications will commence. If there is no communications check the common baud rate, unique station numbers, the wiring, the termination resistors and finally the devices themselves.

The Input/Output definitions need to be matched with PLC data registers (see examples).

The System Area handshake flags need to be serviced by the PLC (see above).

At this stage communications is active and will be displayed by the LED indicators on the CC-Link communications module.

Parameters are either INPUT parameters read by the CC-Link Master or OUTPUT parameters written by the CC-Link Master.

Data Formats

Data is returned as ‘scaled integers’, such that 999.9 is returned or sent as 9999; 12.34 is encoded as 1234. The control program in the CC-Link master must convert the numbers into floating point values if required.

Troubleshooting

No Communications:

- Check the wiring carefully, ensure that the correct terminals have been wired to.
- Check the 'Comms' list in configuration level and, under 'User' check that the parameter 'Ident' is showing CC-Link. If not, the unit may not be fitted with the correct CC-Link communications module or it is not recognised by the EPower controller unit.
- Check that all devices including the CC-Link Master module have the same Baud Rate.
- Check that the 'Address' (Station Number) in the 'Comms' list is correct and unique for the network configuration in use.
- Check that there are no overlaps between Station Numbers taking into account each device's number of "Occupied Stations".
- Ensure that the network is correctly configured and the configuration has been downloaded correctly to the CC-Link Master Module.
- Ensure that the CC-Link Master Module Input and Output Parameter mapping is correctly matched.
- Ensure that the CC-Link Master Module is set up to service the System Area handshake flags.
- Verify that the maximum line length for the baud rate in use is not exceeded (refer to "Maximum Transmission Distance" on page 89 and the CC-Link standard at www.cc-link.org).
- Ensure that the both ends of the CC-Link network trunk line are correctly terminated ([Figure 38](#)).
- If possible, replace a faulty device with a duplicate and retest.

PROFINET

Introduction

PROFINET is the open industrial Ethernet based networking solution for automation. It is similar to PROFIBUS in that it enables distributed IO control from a PLC. PROFINET uses TCP/IP and IT standards, and is, in effect, real-time Ethernet and enables the integration of existing Fieldbus systems like PROFIBUS, DeviceNet, and Interbus, without changes to existing devices.

PROFINET IO was developed for real time (RT) and Isochronous Real Time (IRT) communication with the de-central periphery. The designations RT and IRT merely describe the real-time properties for the communication within PROFINET IO.

There are four stages to setting up a network:

- Physical Wiring, "PROFINET Wiring" on page 99
- Setting up EPower controller units, "Setting Up the EPower Controller Unit" on page 100
- Data exchange mapping, "Data Exchange Mapping" on page 104
- Setting up the master, "Acyclic (Explicit) Messaging" on page 107, "Data Formats" on page 108 and "The GSD File" on page 109

EPower Controller PROFINET Features

- 100Mbit, full duplex operation
- Galvanically isolated bus electronics
- Field pluggable option
- Polled and Explicit I/O messaging connection

PROFINET Wiring

PROFINET capability is provided by an interface board installed within the unit and provides a single RJ45 socket ("PROFINET IO (Single Port)" on page 21) or dual RJ45 port sockets ("PROFINET IO (Dual Port)" on page 22).

The PROFINET port is a 100 Mbit, full duplex operation port and should be connected via an industrial switch to a Master device (eg PLC) with Cat5e (straight through) cable via the standard RJ45 connector (maximum length 100m).

The interconnecting cables should be fitted with plugs provided with an outer metallic shell with the shell connected to the wire screen of the cable. See also "Wiring 3-Wire EIA485" on page 22 for suitable cables. This type of cable must be used to maintain EMC compliance.

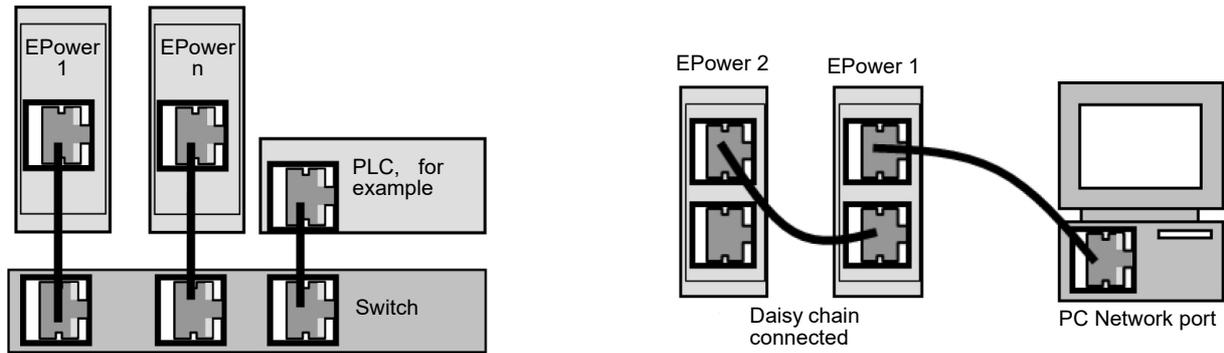


Figure 41 PROFINET Wiring - Multiple EPower controller Units (Single port left, Double port right)

Setting Up the EPower Controller Unit

Eurotherm iTools configuration package connected to the RJ11 configuration port is used to set up parameters in EPower. Further details are available in the EPower User Guide HA179769 and iTools help manual HA028838. It is recommended to setup the unit using the master mode (Comms.PNinitMode = 0) and the configurator tool of your PLC. To avoid conflicts, it is recommended to change the default station name of the unit and use your own station name. This is not essential but network conflicts may occur if the default settings interfere with equipment already on the network.

For the PROFINET instrument the IP address, subnet mask, default gateway and DHCP enable need to be configured. These parameters are available in EPower controller units under different levels of access as detailed in the EPower controller User Guide HA179769.

Changing any one of these parameters may immediately move the instrument to a new network address. For this reason, it is recommended that such changes are made offline.

IP Addresses are usually presented in the form "abc.def.ghi.jkl". In the instrument Comms folder each element of the IP Address is shown and configured separately such that IPAdd1 = abc, IPAddr2 = def, IPAddr3 = ghi and IPAdr4 = jkl. This also applies to the SubNet Mask and Default Gateway IP Address.

Each Ethernet module contains a unique MAC address, normally presented as a 12 digit hexadecimal number in the format "aa-bb-cc-dd-ee-ff".

In EPower controller units MAC addresses are shown as 6 separate hexadecimal values on an EPower instrument itself or decimal values in iTools. MAC1 shows the first address value (aa), MAC2 shows the second address value (bb) and so on.

PROFINET Initialisation Mode (PNinitMode) Parameter

Eurotherm provides several way to initialise the PROFINET communication. These modes are chosen by using the iTools parameter Comms.PNinitMode.

The parameter PNinitMode can take the following values:

0: Master Mode: lets the master decide the PROFINET device name of the EPower as well as its IP address. This is the default value for this parameter. It is recommended to use this value and initialise the station name and the IP address by using the PLC application tool such as Step7.

1: SN IP: Both station name and IP address will be initialised by the value provided in iTools during startup of the EPower. This configuration may lead to errors with some Profinet masters.

2: SN noIP: The station name will be assigned during the Startup of the EPower following the use of the PNDevNum parameter. This configuration may lead to errors with some PROFINET masters.

3: NoSN IP: Only the IP address will be initialised during the start up of the EPower. The station name remains unchanged. This configuration may lead to errors with some PROFINET masters.

Dynamic Host Configuration Protocol (DHCP) Settings

This is set in configuration level by the DHCP Enable parameter.

This is only available if PninitMode=1 (SN IP) or PninitMode=3 (NoSN IP).

It is not recommended to use this with PROFINET.

IP addresses may be 'fixed', set by the user or dynamically allocated by a DHCP server on the network.

If IP Addresses are to be dynamically allocated, the server uses the instrument MAC address to uniquely identify them.

Fixed IP Addressing

This is only available if PninitMode=1 (SN IP) or PninitMode=3 (NoSN IP).

It is not recommended to use this with PROFINET.

In the "Comms" folder of the instrument set the "DHCP enable" parameter to "Fixed". Set the IP address and SubNet Mask as required. This may be done in Engineer level.

Dynamic IP Addressing

This is only available if PninitMode=1 (SN IP) or PninitMode=3 (NoSN IP).

It is not recommended to use this with PROFINET.

In the "Comms" folder of the instrument set the "DHCP enable" parameter to "Dynamic". Once connected to the network and powered, the instrument will acquire its "IP address", "SubNet Mask" and "Default Gateway" from the DHCP Server and display this information within a few seconds.

Note: If the DHCP server does not respond (in common with other Ethernet appliances in this situation) the unit will not be accessible via the network.

Default Gateway

This is only available if PninitMode=1 (SN IP) or PninitMode=3 (NoSN IP).

It is not recommended to use this with PROFINET.

The "Comms" folder also includes configuration settings for "Default Gateway". These parameters will be set automatically when Dynamic IP Addressing is used. When fixed IP addressing is used these settings are only required if the instrument needs to communicate wider than the local area network.

Figure 42 shows the appearance of PROFINET User Comms configuration parameters in iTools.

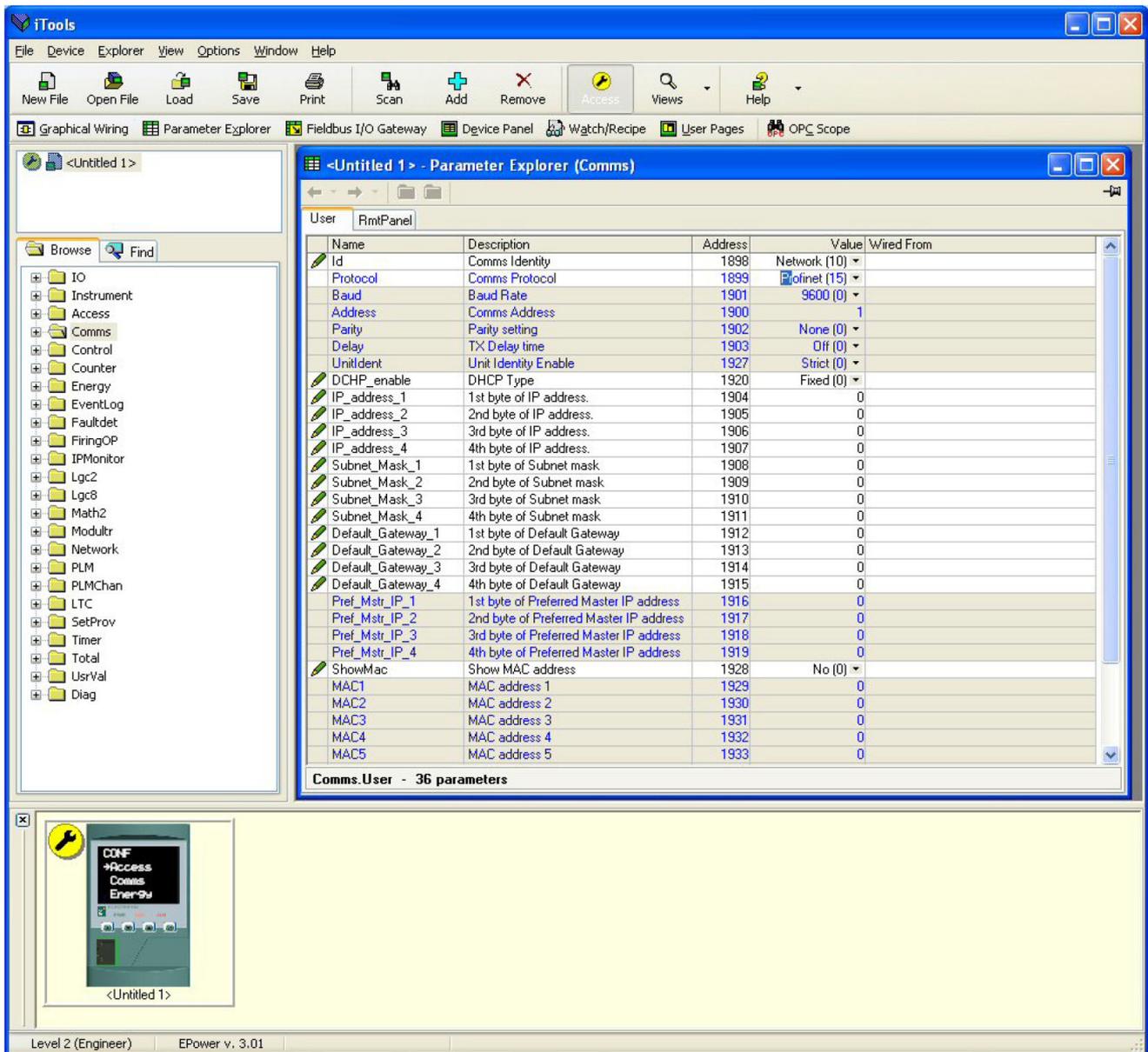


Figure 42 PROFINET Comms Parameters

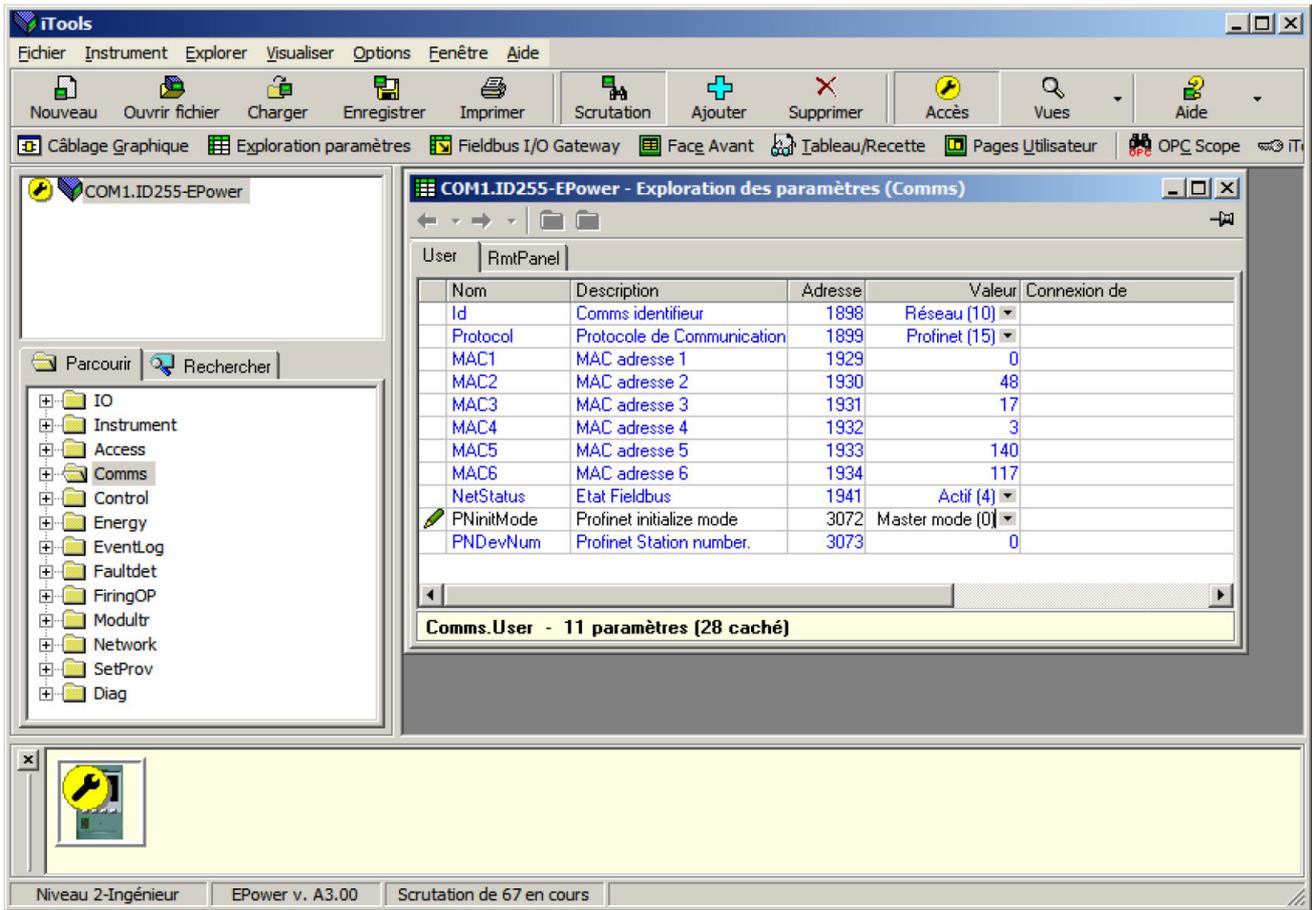


Figure 43 Simplified view of the PROFINET parameters when PNinitmode=0 (master mode).

In this case, the Master will allocate the IP address as well as the station name of the device. This is the recommended mode.

Device Name

Whereas Ethernet uses the MAC address to identify a device without ambiguity, the PROFINET master also uses the Device Name to identify the device. The Device Name must, therefore, be unique over the PROFINET network and allows a device to be replaced or cloned without the need to re-configure the whole system. The Device Name can be a long string up to 240 bytes.

Using iTools, set the parameter 'PNinitMode' to value 1 or 2 (by default this is set to 0'). The value of the parameter 'PNDevNum' (device number) is used to build the station number. Assign a number into this parameter. In this case the station name will be 'EPower.sXXX' where XXX is the number entered into the parameter 'PNDevNum'. The name can be cloned and is synchronised every time the driver module is initialised.

The driver needs to be restarted following any change to these two parameters.

'PNdevnum' can be between 0 and 32000 where 0 is the default value.

Figure 44 shows the implementation of Device name.

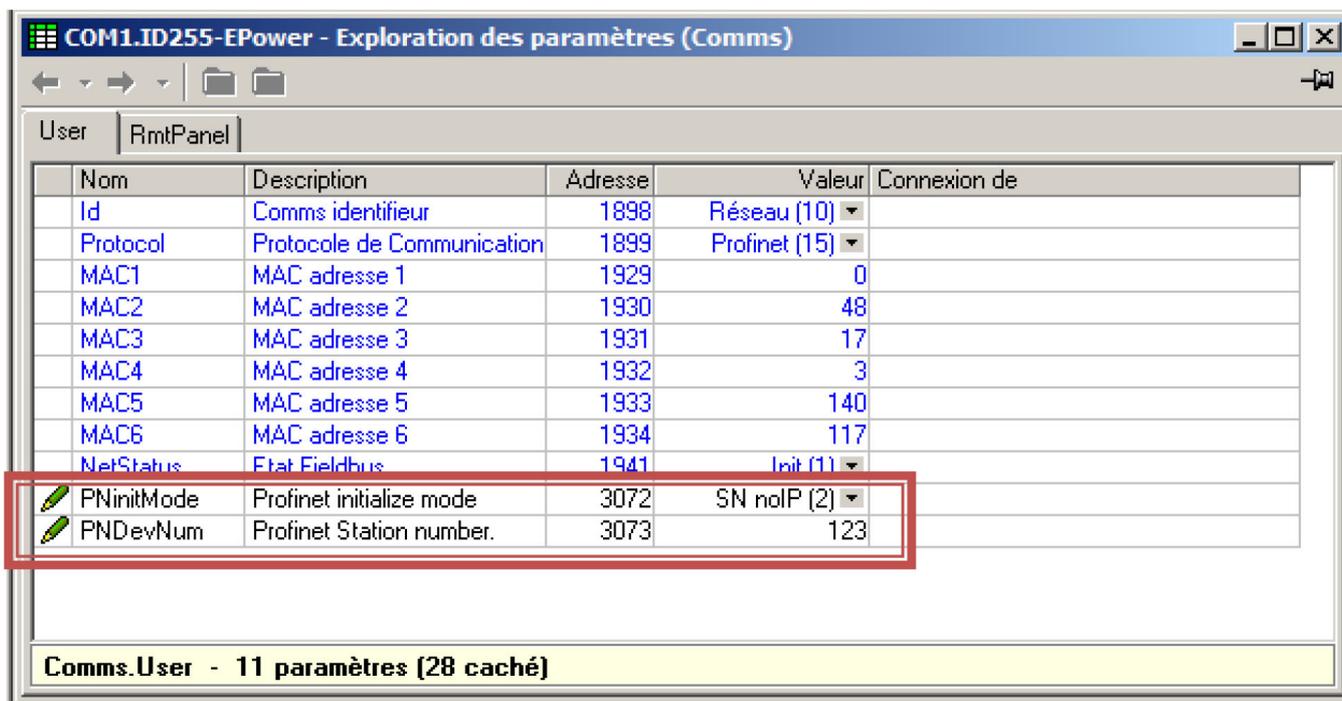


Figure 44 Device name

⚠ WARNING

To change the EPower station name using your configuration tool (like Step7), it is necessary to set the parameter 'COMMS.PNInitMode' to 'MASTER mode' (0). Failure to do this will result in the value set with your configuration tool to be overwritten the next time EPower starts. This will un-configure your network.

Data Exchange Mapping

Up to 32 input and 16 output parameter variables may be included in the PROFINET cyclic (implicit) data exchange.

By default, the most frequently used values are included, but it is possible to select other parameters within the unit. The default mapping is as follows:

Input Parameter	Output Parameter
Main PV (Network 1)	Main Setpoint (Network 1)
Main PV (Network 2)	Main Setpoint (Network 2)
Main PV (Network 3)	Main Setpoint (Network 3)
Main PV (Network 4)	Main Setpoint (Network 4)

Input and Output Parameters are 16 bits (2 bytes) each.

To set up the EPower controller unit so that the desired parameters can be read and written involves setting up the INPUT and OUTPUT data tables. This is carried out using iTools.

Configuring the Cyclic (Implicit) Data Exchange

The PROFINET master may be required to work with many diverse slaves from different manufacturers and with different functions. Furthermore EPower controller units contain many parameters most of which will not be required by the network master for a particular application. It is, therefore, necessary for the user to define which Input and Output parameters are to be available on the PROFINET network. The master may then map the selected device parameters into the PLC input/output registers.

Values from each slave, 'Input Data', are read by the master, which then runs a control program. The master then generates a set of values, 'Output Data', into a pre-defined set of registers to be transmitted to the slaves. This process is called an 'I/O data exchange' and is repeated continuously, to give a cyclical I/O data exchange.

The Input/Output definitions for PROFINET are configured using iTools in the same way as for DeviceNet or Profibus.

Select the 'Fieldbus I/O Gateway' tool from the lower toolbar, and an editor screen will appear similar to that shown in [Figure 45](#).

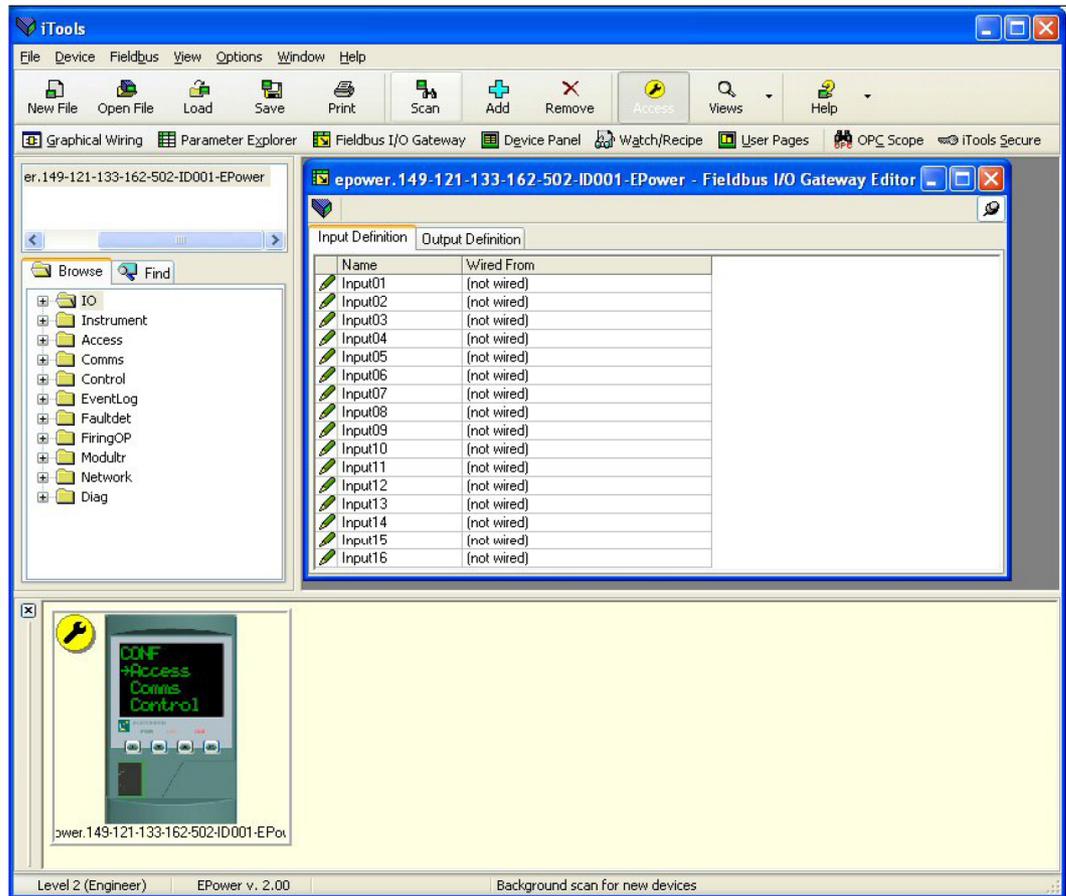


Figure 45 The I/O (Fieldbus I/O Gateway) Editor in iTools

There are two tabs within the editor, one for the definition of 'Inputs', and the other for 'Outputs'. Inputs are values sent from the EPower controller to the PROFINET master, for example, alarm status information or measured values, i.e. they are readable values. 'Outputs' are values received from the master and used by the EPower controller, for example, setpoints written from the master to EPower controller. Note that Outputs are written on every PROFINET cycle, which is frequent, of the order of 10-100mS, and so values from PROFINET will overwrite any changes made on the EPower controller keypad unless special measures are taken to prevent this.

The procedure for selecting variables is the same for both input and output tabs. Double-click the next available position in the input or output data and select the variable to assign to it. A pop-up provides a browser from which a list of parameters can be opened. Double-click the parameter to assign it to the input definition. Note that you should assign inputs and outputs contiguously, as a 'not wired' entry will terminate the list even if there are assignments following it.

Figure 46 shows an example of the pop-up and the input list produced.

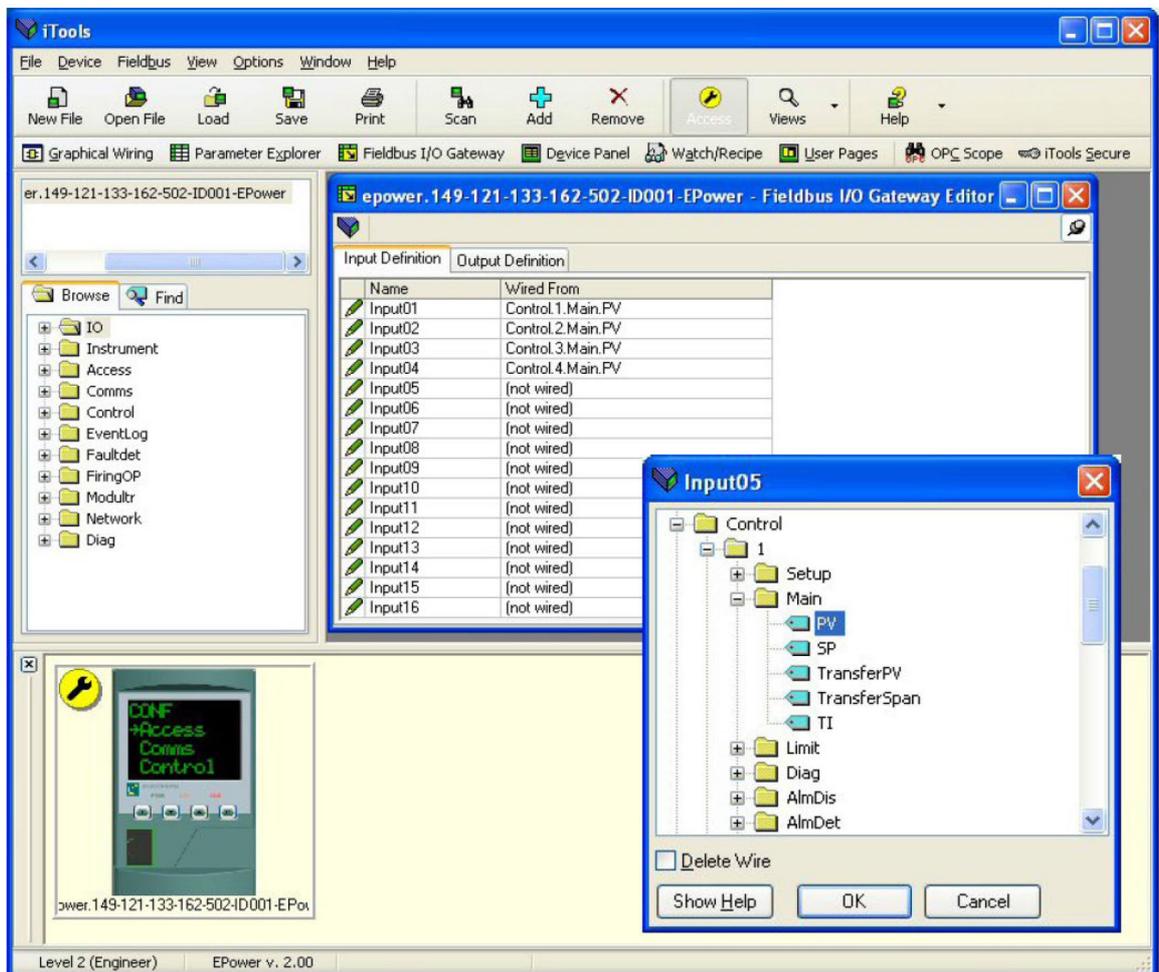


Figure 46 Selecting an Input Value and Example of an Input List

When the list is populated with the variables you require, note how many 'wired' entries are included in the input and output areas as this will be needed when setting up the PROFINET Master. In the example above, there are four input values, each of two bytes in length, so a total of 8 bytes of data. Note this number, as it is required when setting the I/O length when configuring the PROFINET master.

Note: No checks are made that output variables are writeable, and if a read only variable is included in the output list, any values sent to it over PROFINET will be ignored with no error indication.

Once the changes have been made to the I/O lists, they must be downloaded to the EPower controller unit.

This is done with the button on the top left of the I/O Editor marked .

The EPower controller Unit will need to be powered off and on again once this has been done for the changes to register.

The next step in the process is to configure the PROFINET master.

Acyclic (Explicit) Messaging

Acyclic (or explicit) messaging is used to transfer data that does not require continuous updates.

It is possible to access any parameter in the EPower controller unit by means of 'explicit messaging', regardless of whether it has been included in the PROFINET input/output data assembly. To do this, it is necessary to configure an explicit connection in the PROFINET master.

To access parameters, use the ADI object (Class 0xA2 hex), using an instance number equal to the Modbus address of the parameter.

A list of Modbus addresses is given in the EPower controller User Guide or can be accessed using iTools as shown in [Figure 47](#).

PROFINET Acyclic Readings

This section describes how to access a variable using PROFINET in acyclic mode.

PROFINET uses the following parameter to access a variable in acyclic mode:

- API
- Slot and Subslot
- Index

To access a parameter in acyclic mode, you first need to know its Modbus address. This may be accessed by selecting the parameter from the browse list as shown in "iTools" on page 25. [Figure 47](#) shows an alternative way to access a parameter. This uses the Graphical wiring editor. The Modbus address is shown in Address column. Right click on the parameter to open the parameter help window.

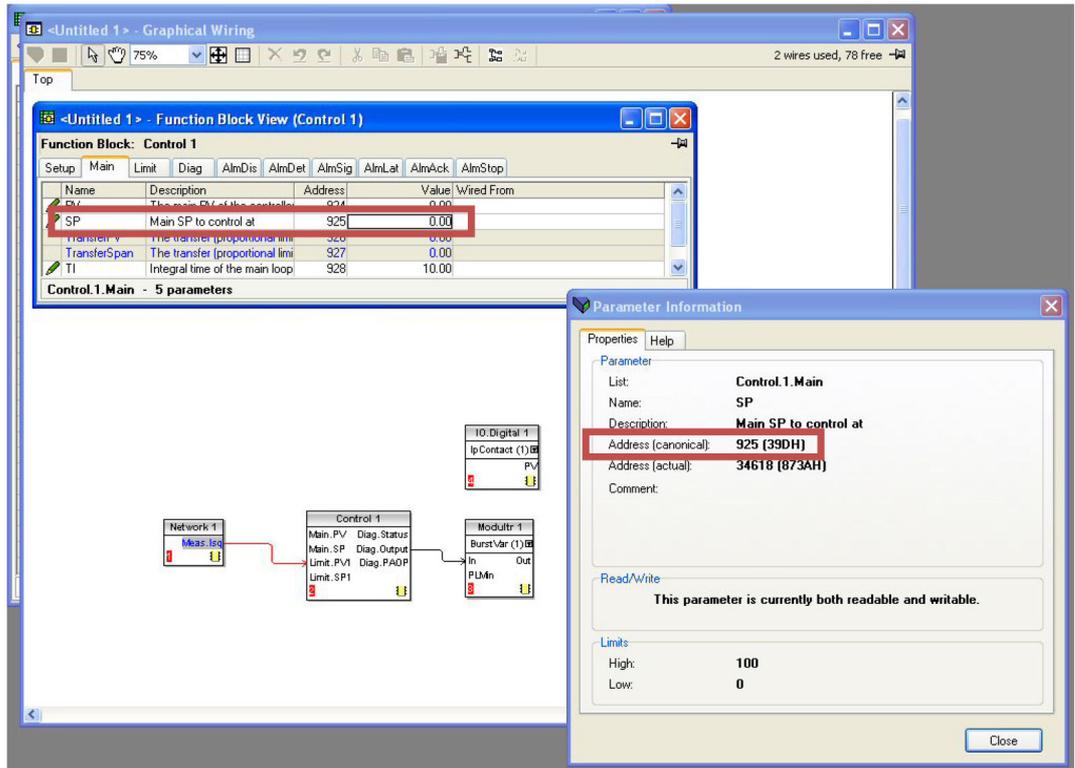


Figure 47 Locating the Modbus Address in iTools

From this address, use the following conversion to get the PROFINET way of addressing a parameter:

- The API is always 0 (Zero)
- The Slot is always 0 (Zero)
- The Subslot is always 1 (one)
- The Index will be the Modbus-address you found previously in iTools

⚠ WARNING

From V3.01: When writing to parameters which are stored in non-volatile memory using cyclic mode, the value is not stored in non-volatile memory to avoid damaging it. Be careful, do not reproduce cyclic mode in PLC by using acyclic writing in a continuous loop. This will permanently damage the EPower driver.

Constraints on the Parameters

The parameter in acyclic-mode follows the same limitation as the parameters in the Fieldbus I/O gateway: 16 bits length and they follow the same scaling, see "Configuring the Data Exchange" on page 62 or "Data Exchange Mapping" on page 104.

Data Formats

Data is returned as 'scaled integers', such that 999.9 is returned or sent as 9999; 12.34 is encoded as 1234. The control program in the PROFINET master must convert the numbers into floating point values if required.

The GSD File

The PROFINET GSDML (General Stations Description) file for EPower controller is named:

- GSDML-V2.2-HMS-ABCC-PRT-20090618.xml, single port version ("PROFINET IO (Single Port)" on page 21)
- GSDML-V2.3-HMS-ABCC-PRT2P-20140703.xml, dual port version ("PROFINET IO (Dual Port)" on page 22)

Which is available from your supplier, or electronically by going to Web site (www.eurotherm.com).

It will also be available where the upgrade tool has been installed (for example, in `c:\Program Files\Instrument Upgrade EPower V3.04`).

The GSD file is designed to automate the PROFINET network configuration process by precisely defining the required device parameter information. Software configuration tools utilise the GSD file to configure an PROFINET network.

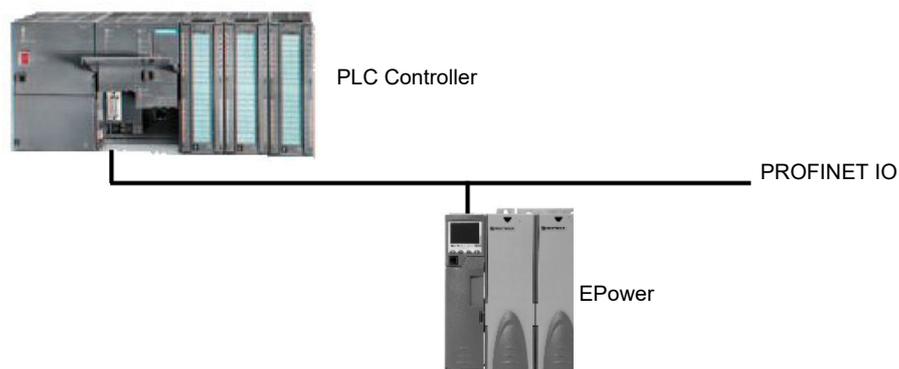
Example - Using A PLC to Configure Epower as a PROFINET I/O Device

The purpose of this section is to provide practical assistance to help set up an EPower master on a PROFINET network. Specific hardware has to be used and this example uses a Siemens Step7 PLC.

Requirements

- Siemens Step7 v5.4 v service pack 5
- Siemens PLC Programming Cable
- PC with Siemens PLC programming software
- EPower with PROFINET communication board
- iTools V7.60 or higher with V3.04 IDM installed
- GSD file for the EPower-controller (shipped within the upgrade-tool from V3.04)

Solution overview



Information about the Ethernet Configuration

For the PROFINET configuration, most of the problem will come from the Ethernet configuration.

To setup a PROFINET configuration, you will need the following information:

- IP address of your PLC
- IP address of your EPower device
- Mac address of your EPower device
- Device name of your EPower device

This information may be found from the following devices:

Item	Read from iTools	Write from iTools	R/W from PLC or configuration tool
IP address of the PLC	No	No	Yes
IP address of the EPower device	Yes	Yes	Yes (DCP)
Mac address of the EPower device	Yes	No	Read only
Device name of the EPower device	Yes	Yes	Yes

Each node on a PROFINET network must have a unique IP address, see also "Setting Up the EPower Controller Unit" on page 100. and a unique Device Name, see also "Device Name" on page 103.

The MAC address of an Ethernet-capable device (like a PROFINET device) is unique by definition.

The Ethernet configuration used in this example is shown in the table below:

Element	Value	Further information
IP address of the EPower PROFINET IO interface	192.168.0.2 (this is 192.168.111.222 by default)	page 100
Subnet mask	255.255.255.0	page 100
MAC address of EPower's PROFINET IO device	00-30-11-03-8C-6F (hex) 00-48-17-3-140-111 (decimal)	page 100
Device name of EPower's PROFINET IO device	eurotherm.epower.station.s1 (epower.s0 is the default station name)	page 103
IP of the PLC	192.168.0.1	Refer to your PLC documentation

PLC Configuration

In this example, the PLC system hardware configuration is done solely with the Siemens Step7 tool.

In order to configure the bus it is necessary to set up the PLC and Controller hardware first. In this example an S7315-2 CPU and a 2A power supply with a standard rack is used. Start the Simatic software and create a new project.

The Welcome screen is shown when the 'Simatic' software is started. Click 'Cancel' to close the Wizard.

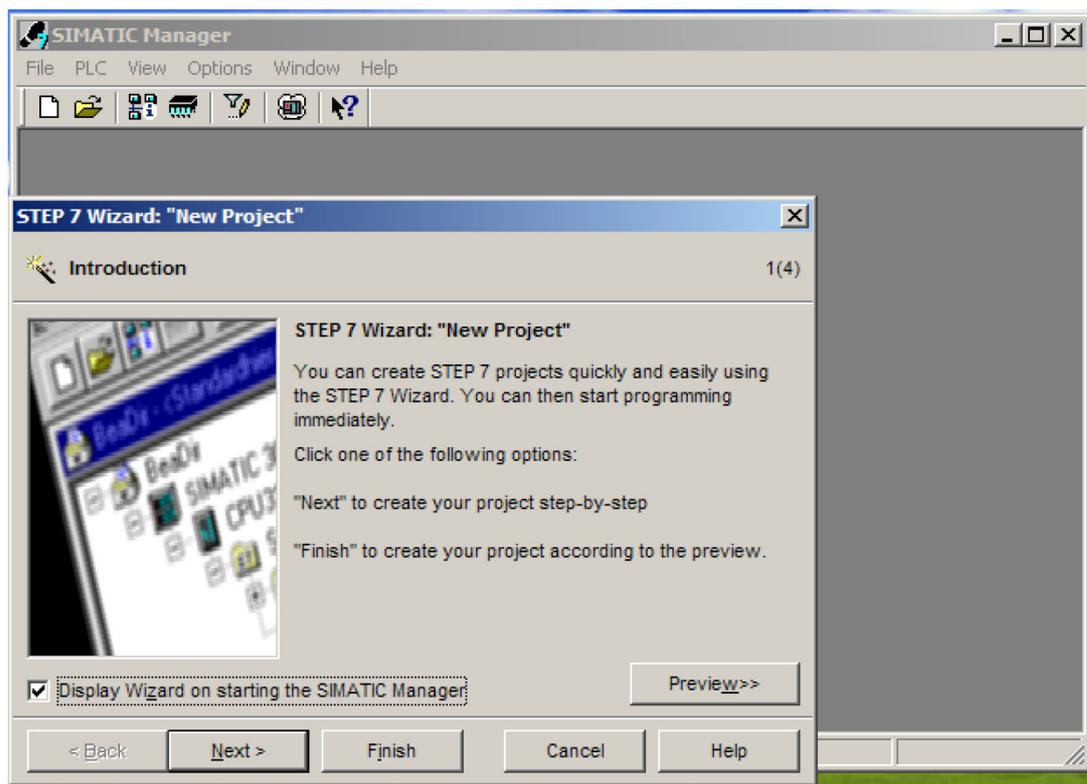


Figure 48 Welcome Screen

Click on File->New, the 'New Project' window appears. Enter a project name and click on 'OK'

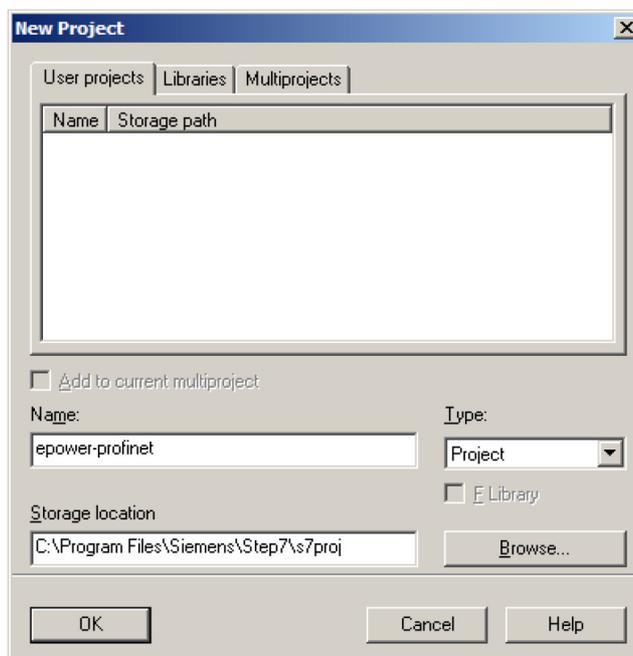


Figure 49 New Project Window

Insert a PLC into the Project

Right click on the project name and insert a Simatic 300 Station as shown in [Figure 50](#).

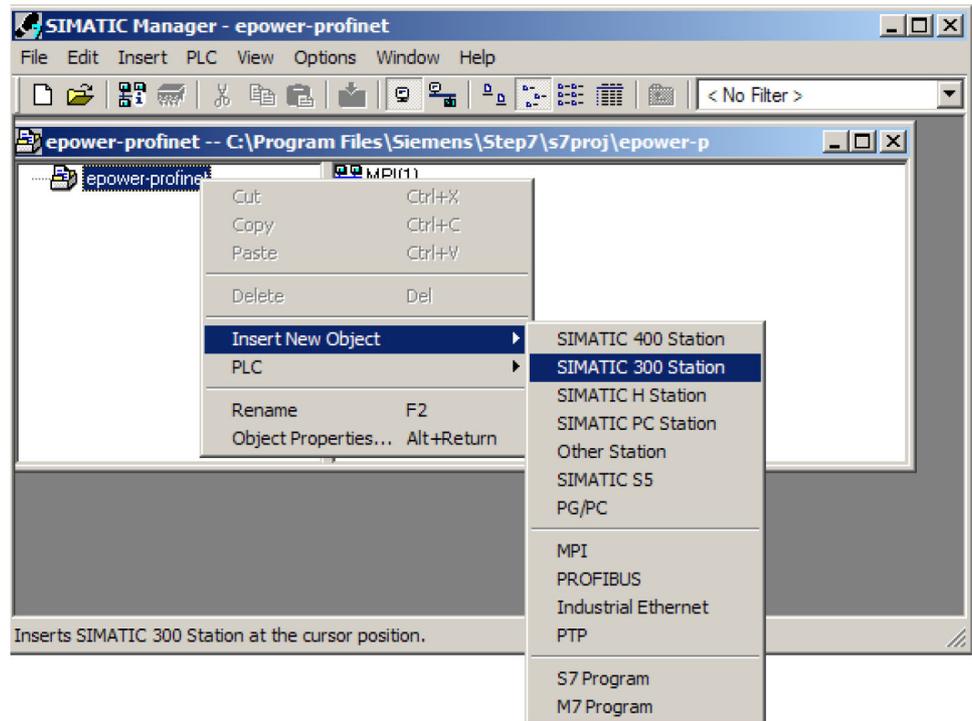


Figure 50 Insert a Siemens PLC in the Project

Double click on the new SIMATIC 300 station then on Hardware to open the hardware configuration.

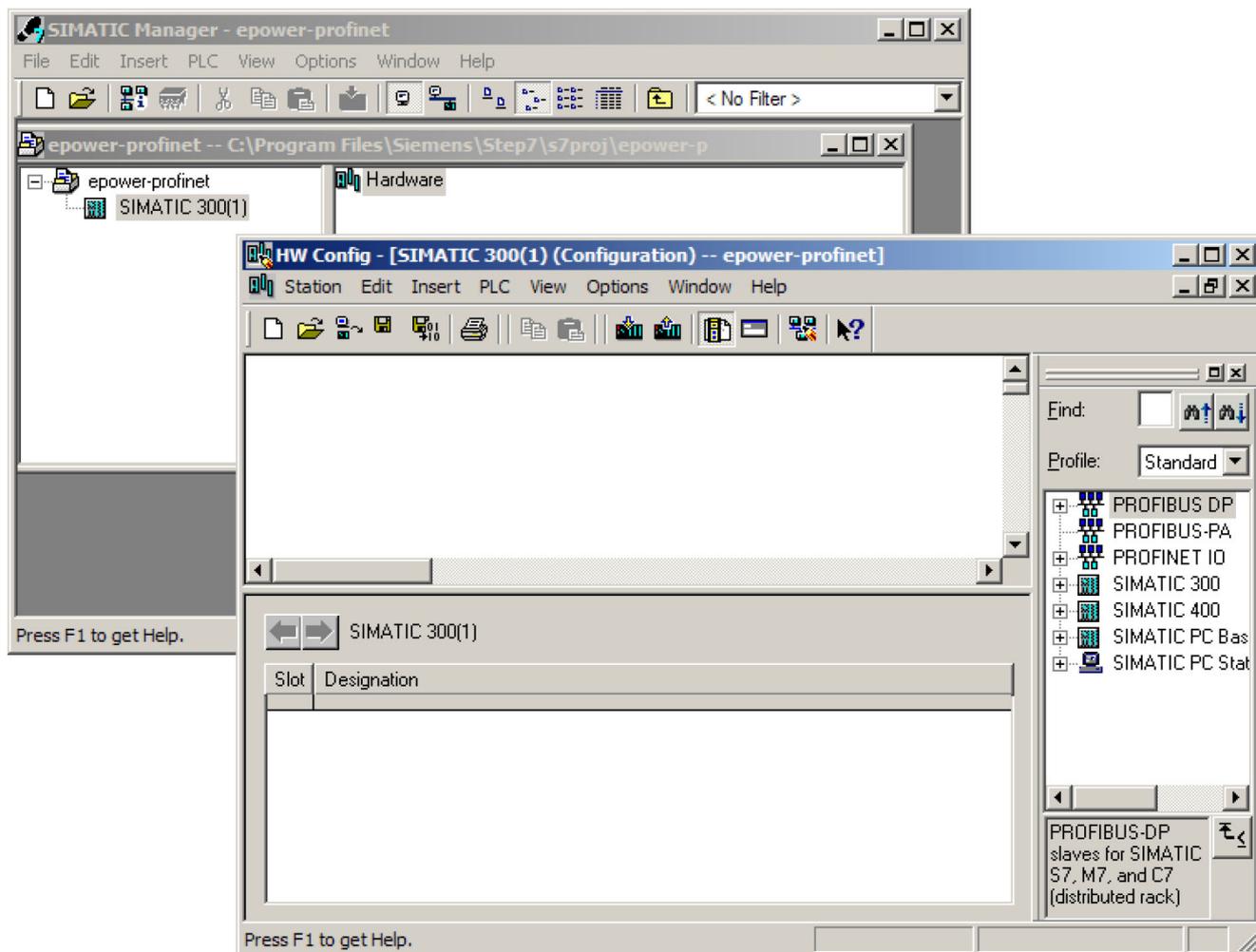
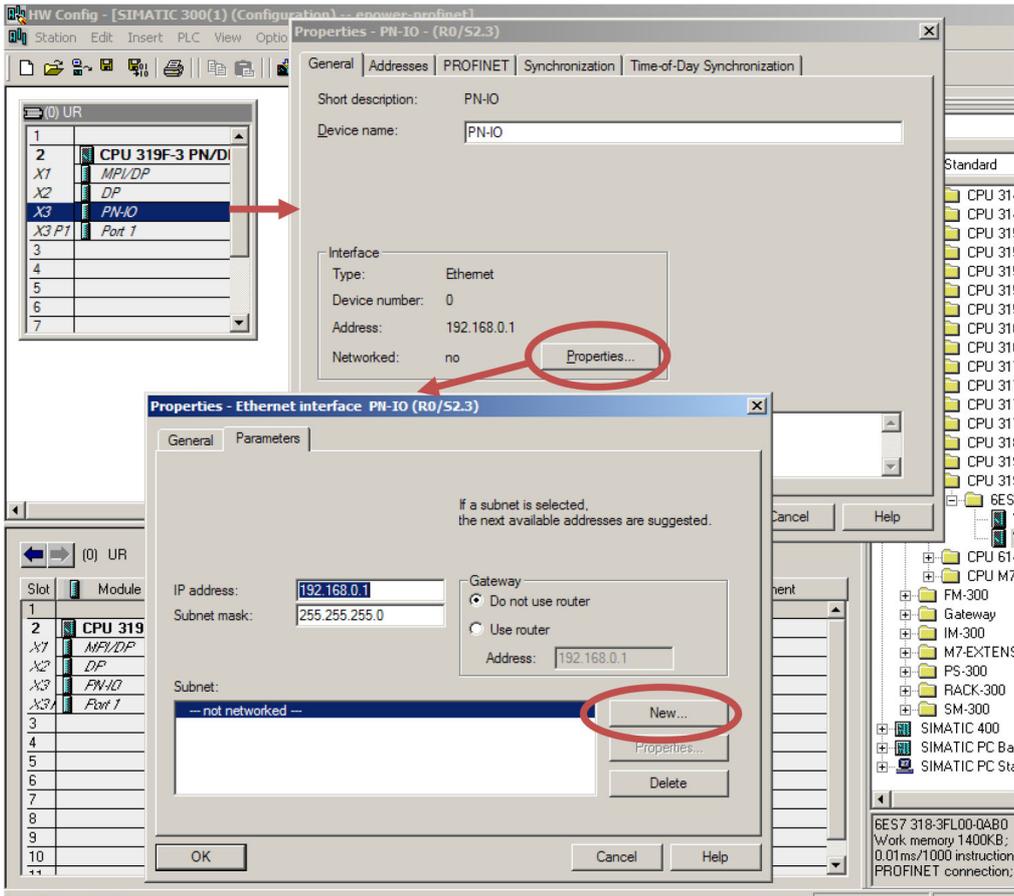


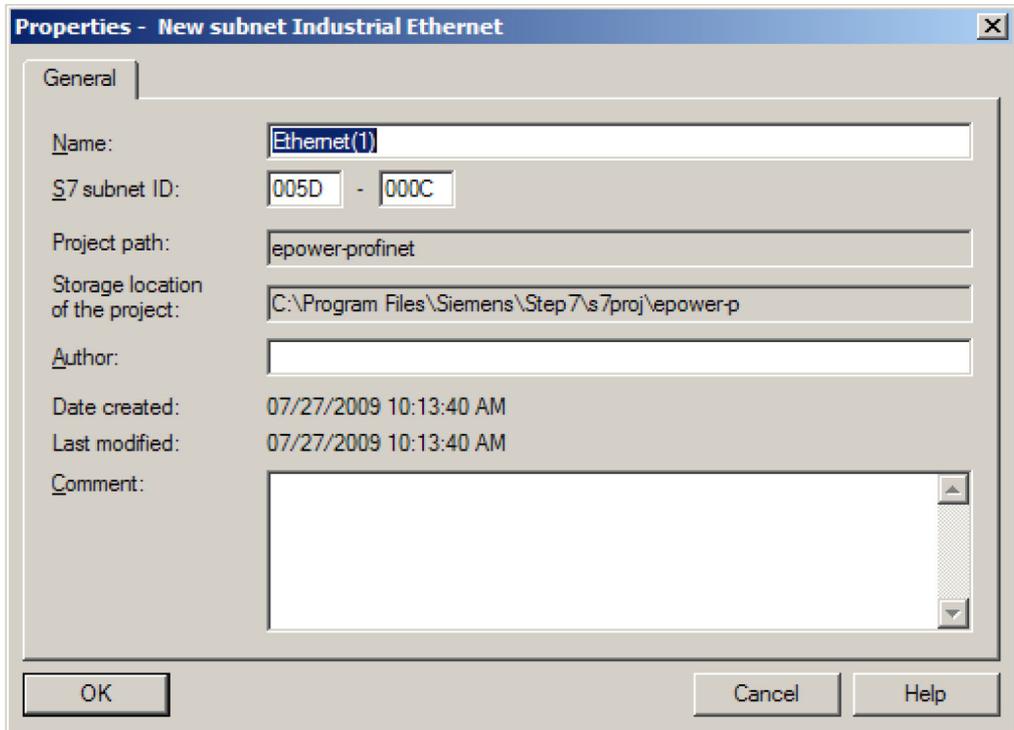
Figure 51 Empty Workspace

Add a Rail, the Power Module, the PLC and the PROFINET Module.

Double click on the PN-IO, PROFINET IO Controller module, to configure the PROFINET IO network. Click on properties in the dialogue, select the desired settings and press OK as shown below:

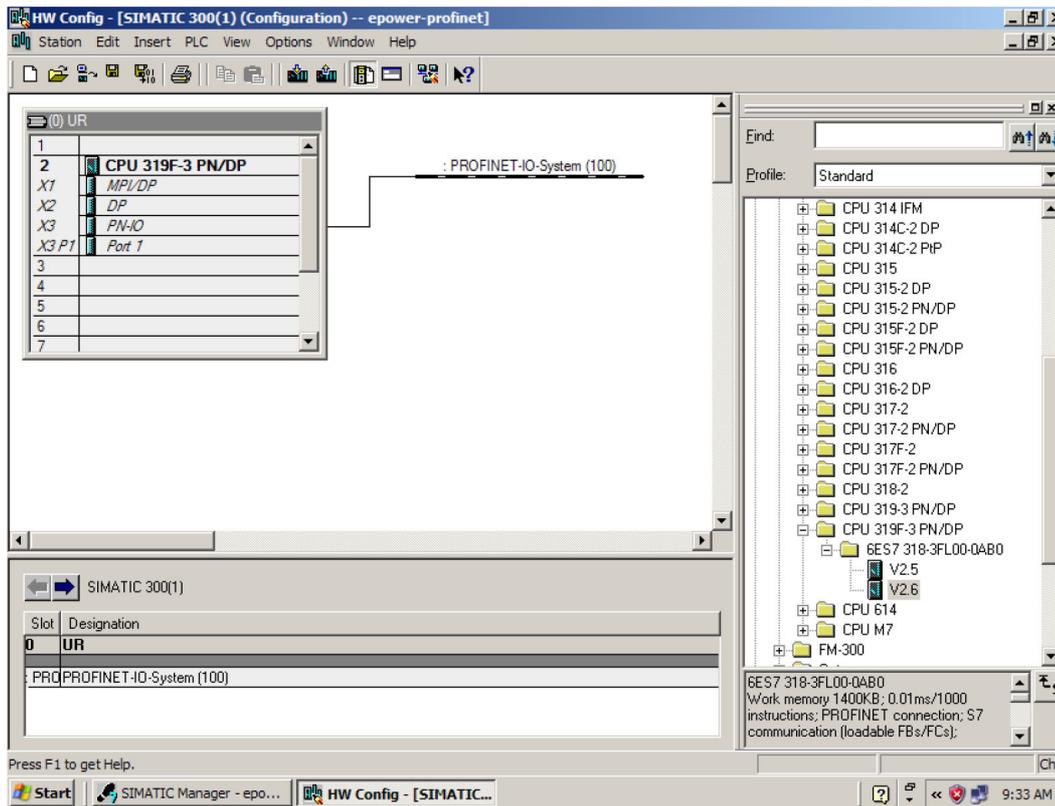


Changing the properties of the PROFINET IO module and defining a new PROFINET IO network. An IP address of 192.168.0.1 and a subnet mask of 255.255.255.0 is used in the configuration described in this example.



Configuring the properties of the PROFINET network.

When the PLC hardware is set up, the screen view shown below should be seen:



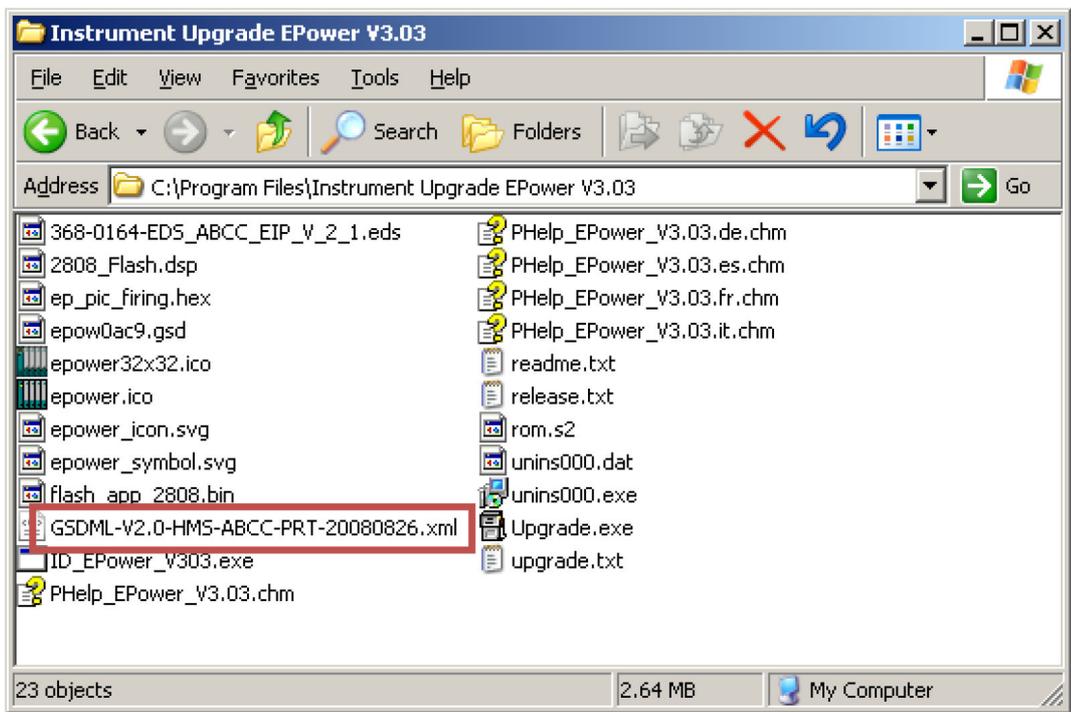
Workspace once the PLC/PROFINET network has been added. PLC with empty bus

Figure 52 Workspace for the PLC/PROFINET Network

STEP7 First-time Configuration – Install the GSD File

If EPower is configured with PROFINET for the first time, you will need to import the GSD file for EPower into Step7.

Firstly, locate the GSD file shipped with your EPower instrument. This GSD file is shipped within the “upgrade tool” from the revision V3.04. Note, Only V3.04 or higher supports PROFINET - this section is given as an example of how to locate the GSD files.



The upgrade tool is usually installed in the directory "C:\Program Files\Instrument Upgrade EPower VX.XX", where X.XX corresponds to current version of the upgrade tool. (In practice this will be V3.03 or newer for your device).
 The GSD file for EPower is called
 GSDML-V2.2-HMS-ABC
 C-PRT-20090618.xml

Figure 53 GSD File Shipped with Upgrade Tool

Once the GSD file is located, it is time to import it into Step7. In the "Hardware configuration" window, open the "Option->Install GSD file" Menu as shown below. This choice imports a new GSD file.

Note: It may be necessary to close the active project first to be able to perform the GSD import.

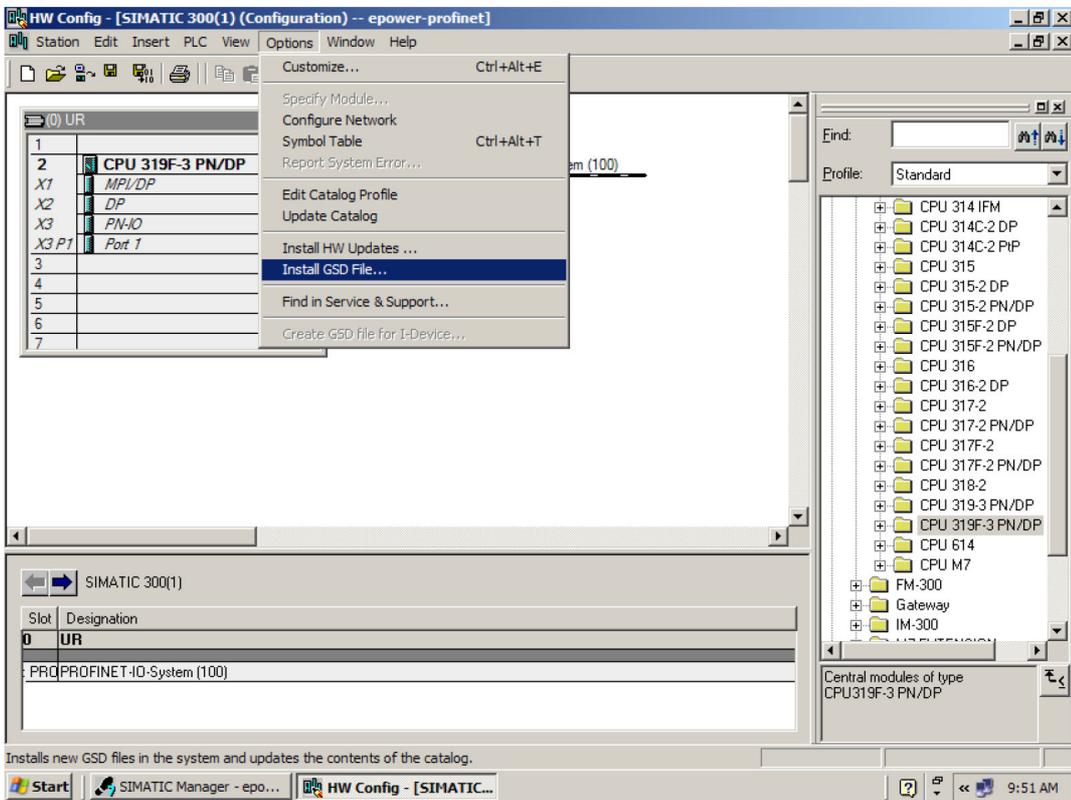


Figure 54 Import EPower GSD into Step7

Click on “browse” and locate the directory where you previously installed the upgrade tool containing the GSD file and click OK.

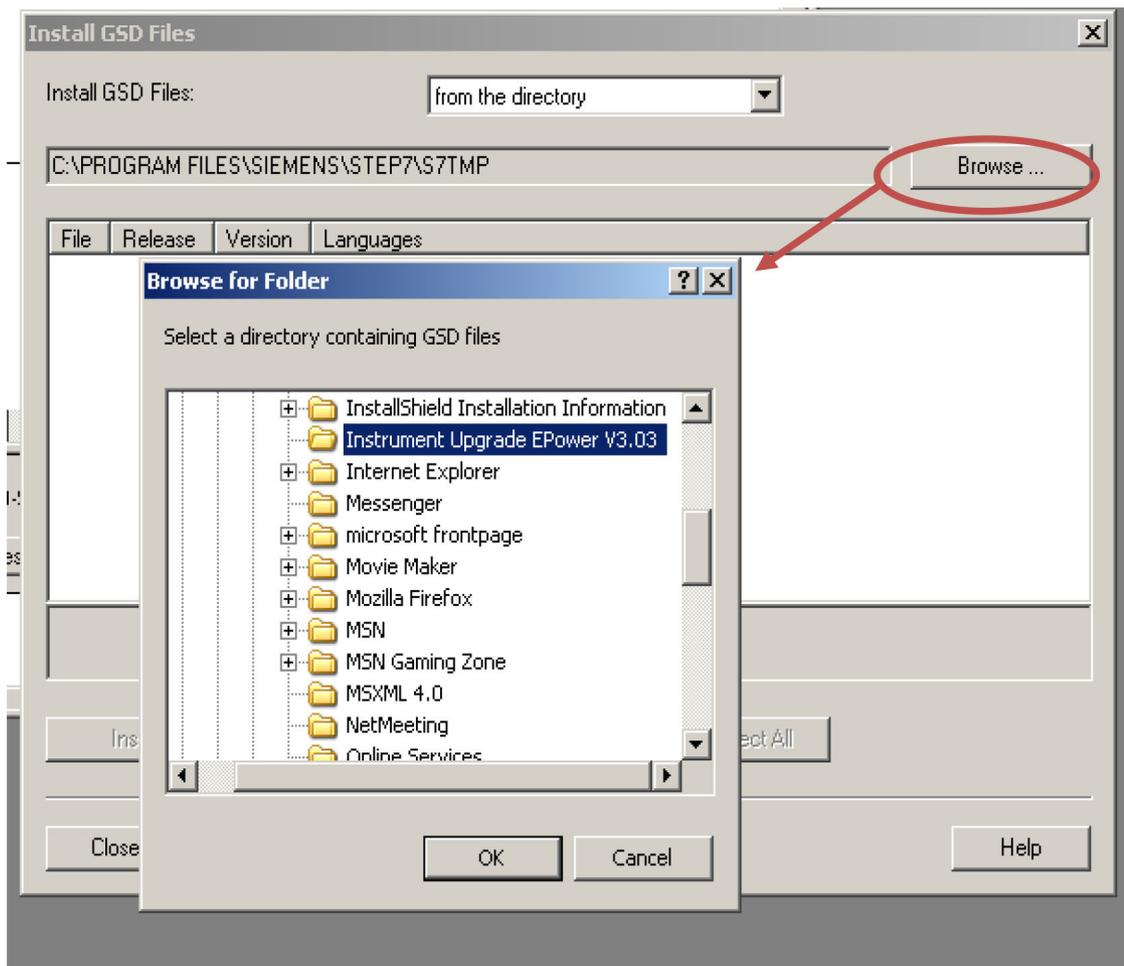
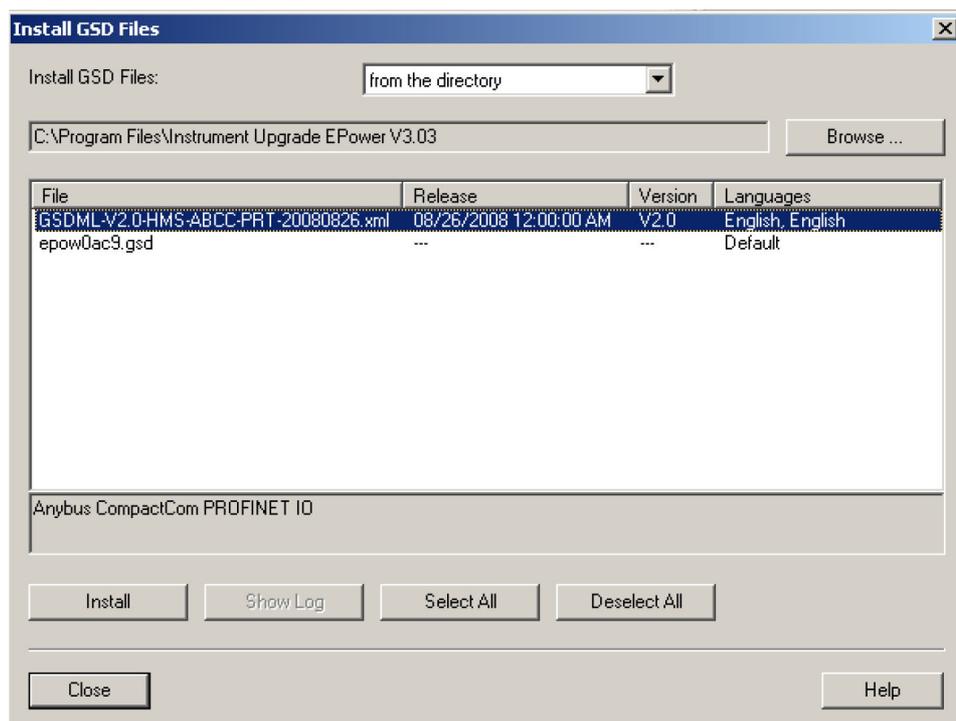


Figure 55 Directory Containing GSD File

This opens the window shown below:



Select the gsdml GSDML-V2.0-HMS-ABCC-PRT-20080826.xml and click on “Install”. Once the GSD file is successfully installed, click on “Close” to close the “Install GSD file” window.

Figure 56 List of Installable Items

Add the EPower Device to the Configuration

From the Treeview “Profinet IO->Additional Field Devices->General->Anybus CompactCom PRT, drag and drop the RT device into the workspace:

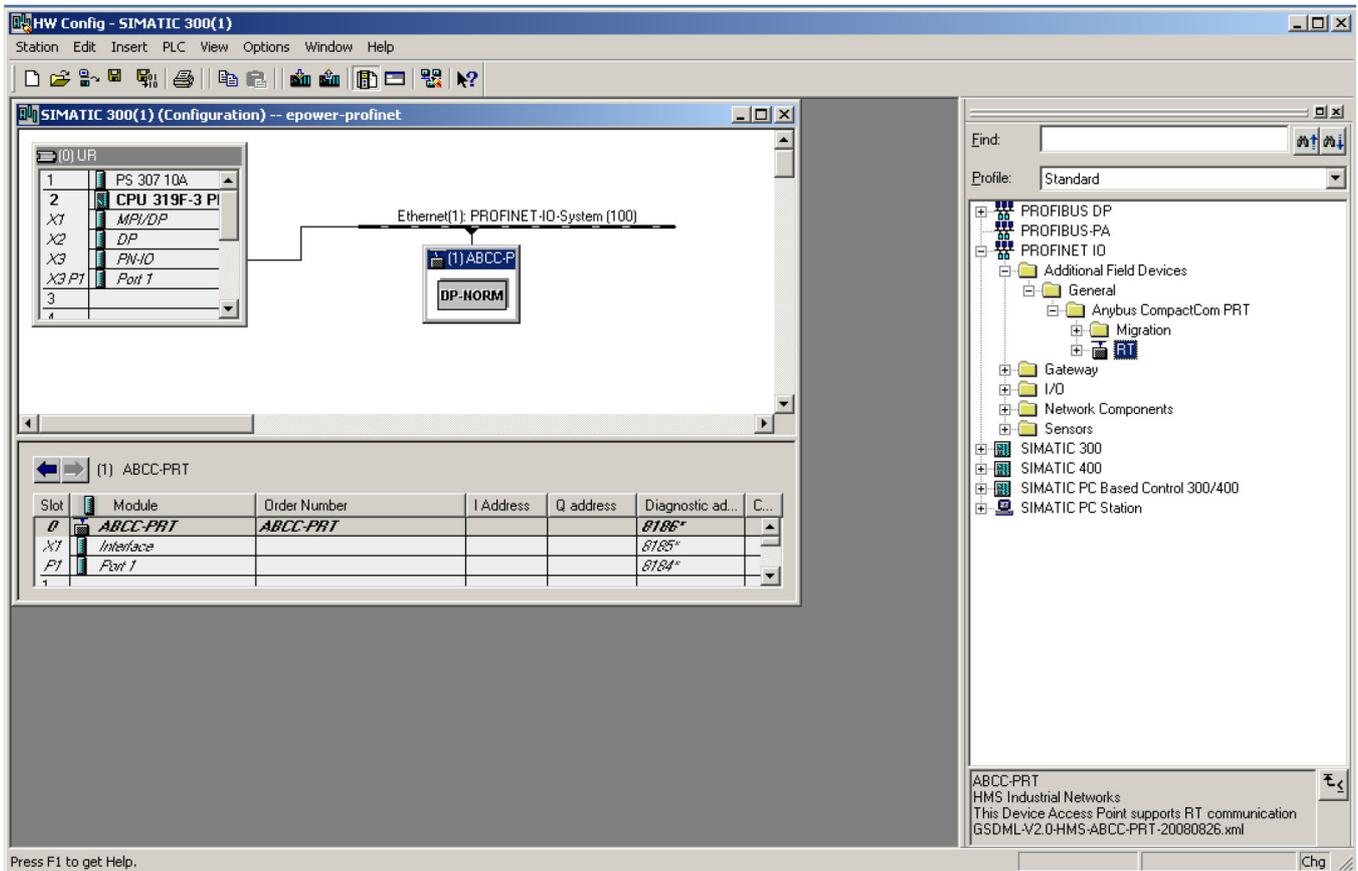


Figure 57 Inserting an EPower Device into the Workspace

Configure the IP Address and the Device Name

To assign the Device Name open the PLC menu and select the Edit Ethernet Node function as shown below:

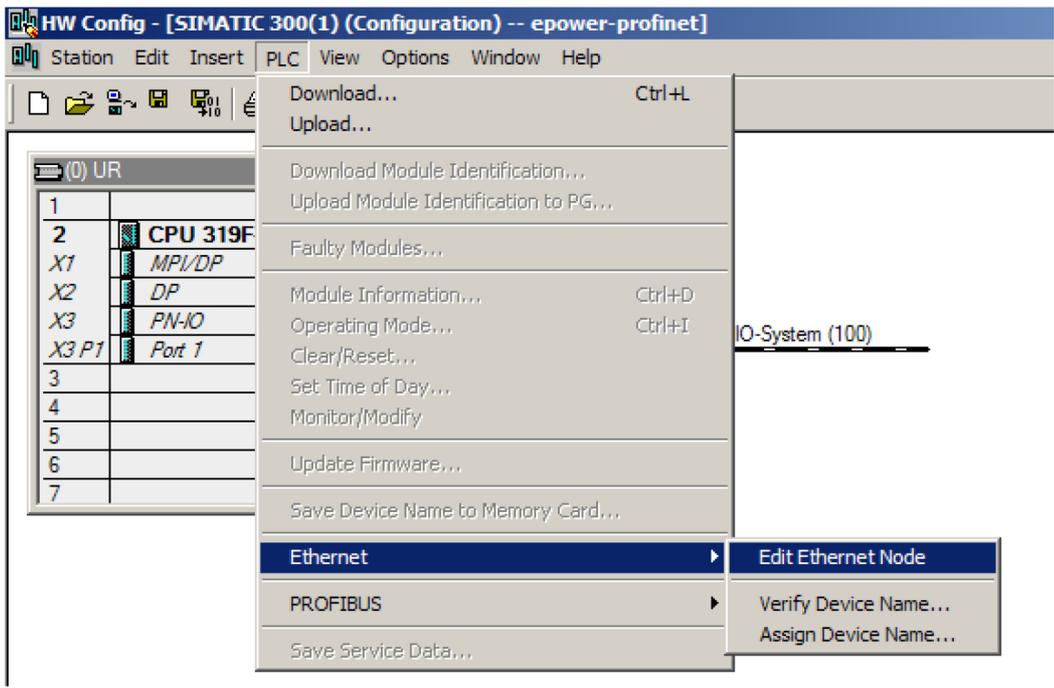


Figure 58 Opening the Ethernet/Profinet Configuration Page

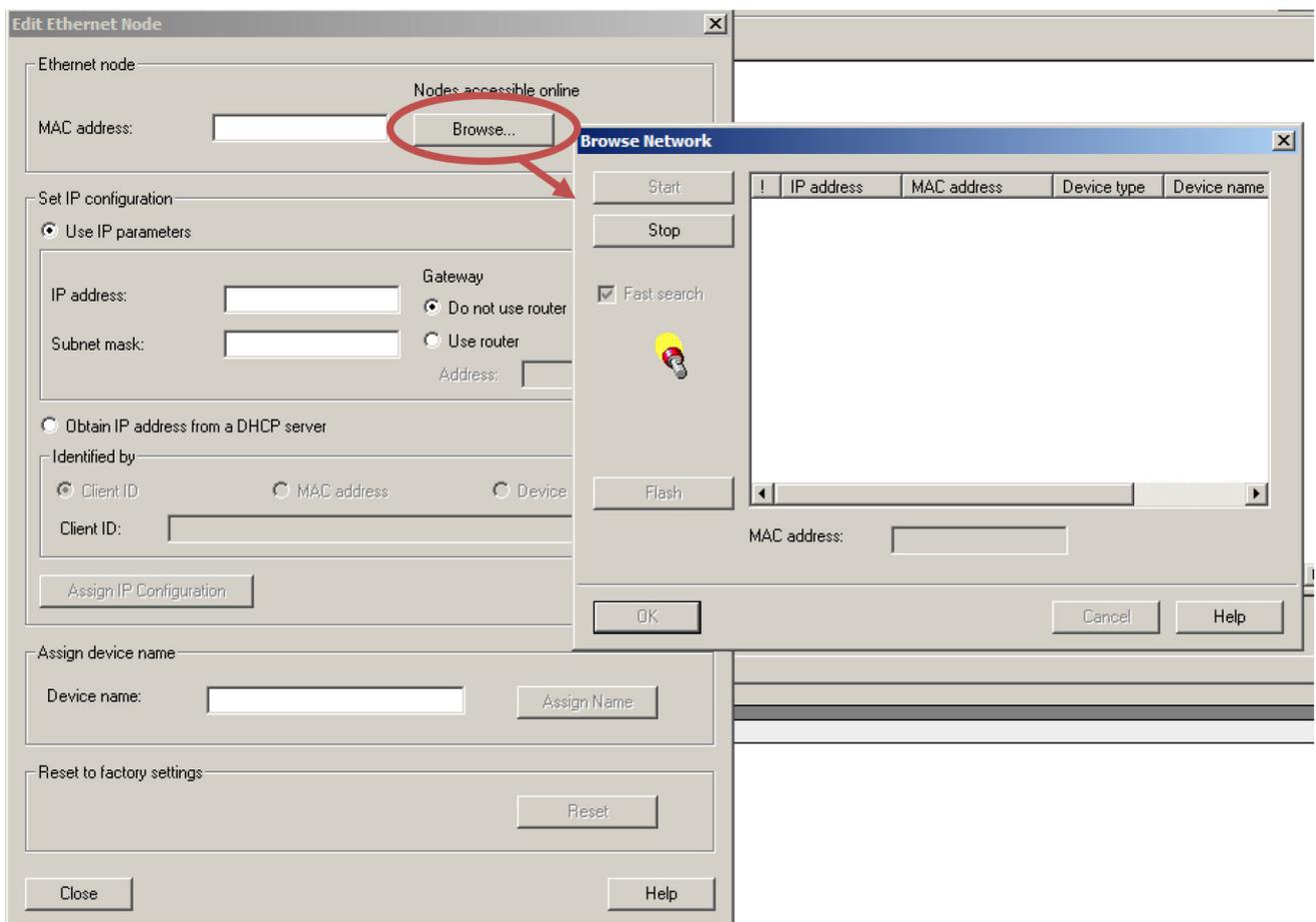


Figure 59 Step7 'Browse Network' Window

After a few seconds, your EPower should be found (see [Figure 60](#)). Notice the IP address which is the same as the IP you have set in iTools previously, and the MAC address which confirms you that you are talking to the right instrument (remember, the MAC address is unique). The current device name is the device name by default (epower.s0).

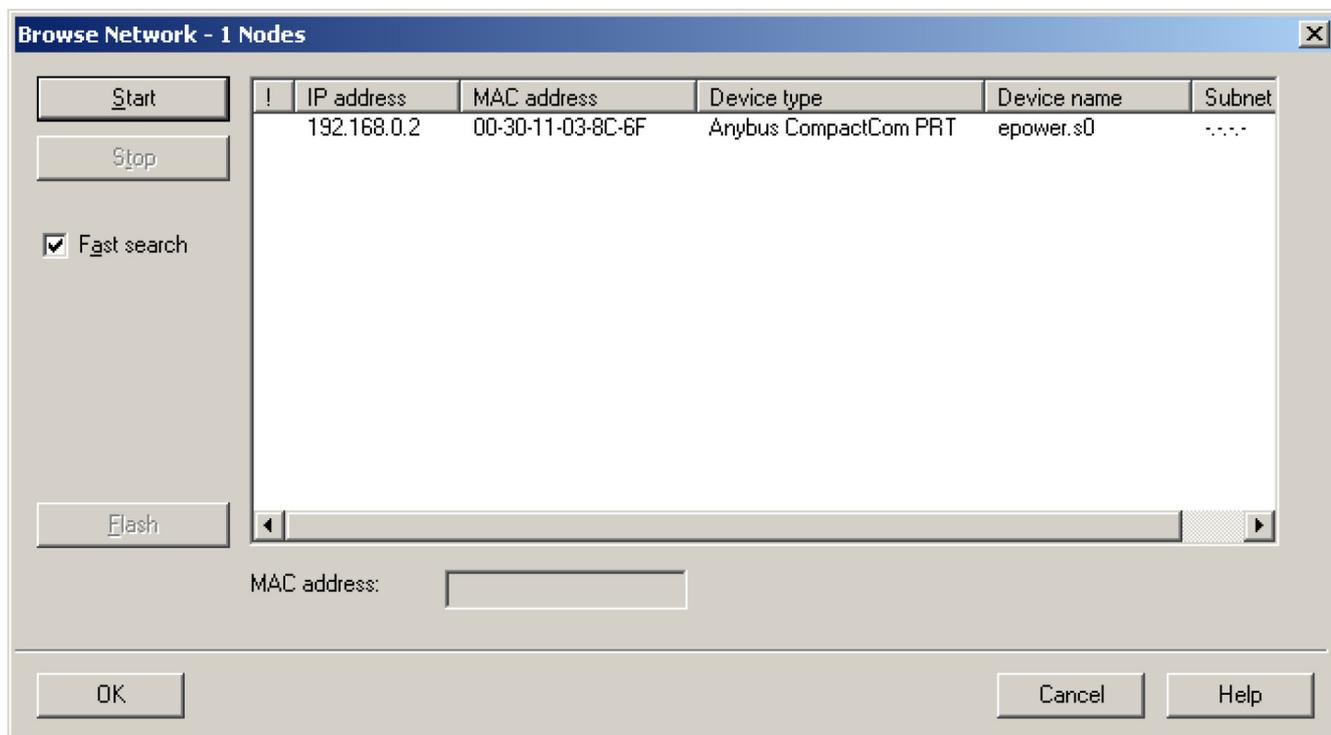


Figure 60 EPower has been found by the Configuration Tool

The device type is identified as a “Anybus CompactCom PRT”. You will recognize here the default station name ‘epower.s0’

Now, select your device and click on the “OK” button.

In the “Edit Ethernet Node” window (below), you can change the IP address of the EPower, and change the “Profinet Device Name” of the device.

In this case, we will change the default station name “epower.s0” with “eurotherm.epower.station.s1”

WARNING

To change the EPower station name using your configuration tool (like Step7), it is necessary to set the parameter ‘COMMS.PNInitMode’ to ‘MASTER mode’ (0). Failure to do this will result in the value set with your configuration tool to be overwritten the next time EPower starts. This will un-configure your network.

Edit Ethernet Node

Ethernet node

MAC address:

Nodes accessible online

Set IP configuration

Use IP parameters

IP address: Gateway

Subnet mask: Do not use router

Use router

Address:

Obtain IP address from a DHCP server

Identified by

Client ID MAC address Device name

Client ID:

Assign device name

Device name:

Reset to factory settings

Figure 61 Configuration of the Unit

To Configure the Application

Once you have configured the “Device Name”, you must enter this same name into your application. To achieve this, double-click on the “EPower” device on the Ethernet bus, and enter the device name, which was written in your real instrument, in the field “Device Name” (see [Figure 62](#)) (we will set “eurotherm.epower.station.s1”).

Secondly, the recommendation is to let the IO Controller set the IP address, but it is also possible to configure the IP address manually by unchecking the box “Assign IP address via IO controller”.

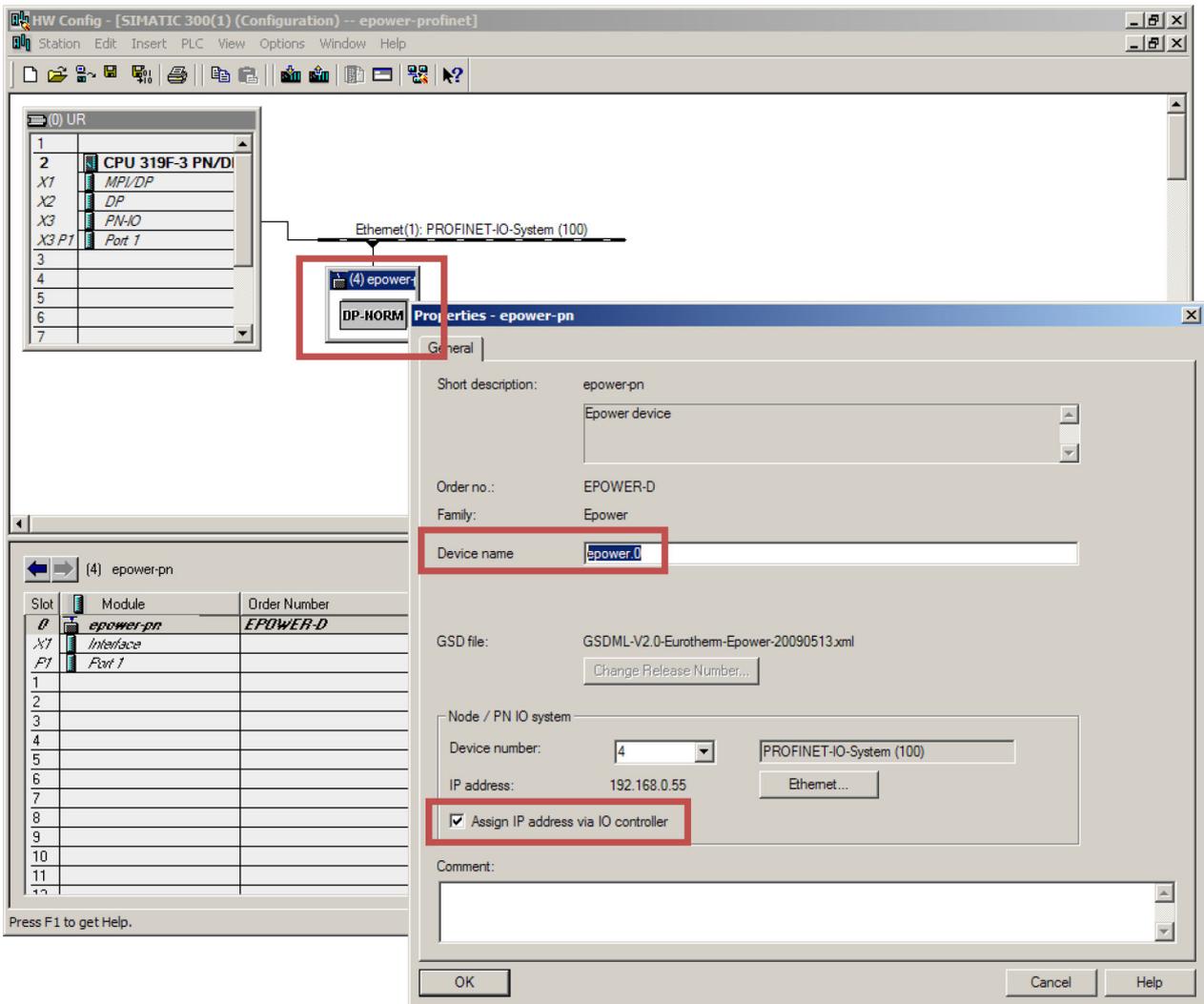


Figure 62 Configuring the Device Name in your Application

I/O Configuration

The I/O configuration for cyclic data exchange must be configured now. The cyclic data exchange follows the rules and limitation of the EPower’s “Fieldbus I/O Gateway” as described in "Configuring the Data Exchange" on page 62 or "Data Exchange Mapping" on page 104.

To allow the application to work, the configuration in the EPower’s “Fieldbus I/O Gateway” and Step7 must match (Figure 62 and Figure 59).

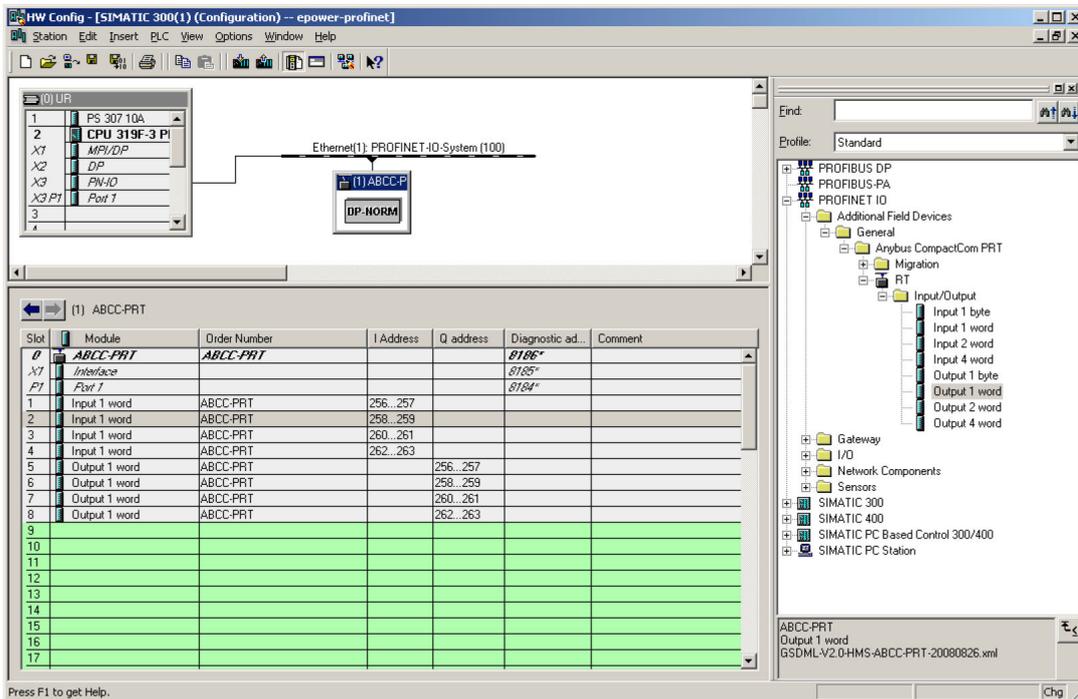
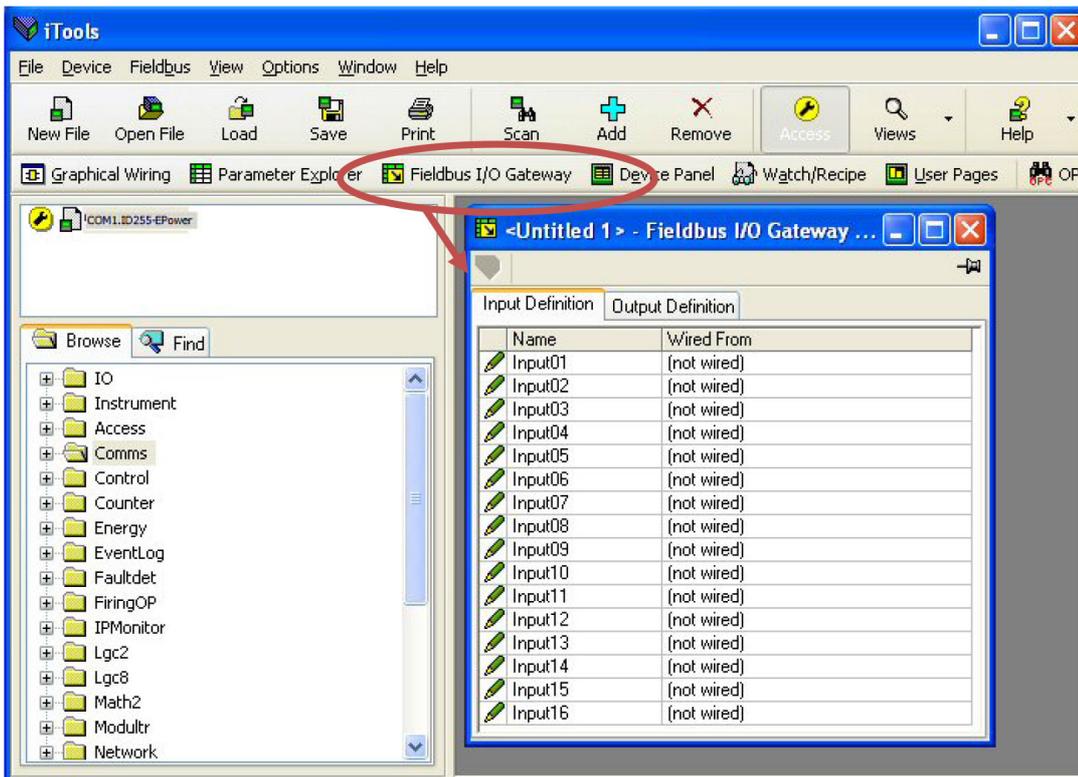


Figure 63 Configuration of the Cyclic Data Exchange in Step7

⚠ WARNING

Despite what is indicated in the list "Input/Output", only the "Input 1 Word" and "Output 1 Word" Data type are allowed for cyclic exchange with the EPower unit. Using other types will lead to configuration error and non working communication. Start to populate the inputs first, then the outputs.

Troubleshooting

No Communications:

- Check the cabling carefully, ensure that PROFINET plugs are fully located in the sockets.
- Check the 'Comms' list in configuration level and, under 'User' check that the parameter 'Ident' is showing Network and the 'Protocol' is showing PROFINET. If not, the unit may not be fitted with the correct PROFINET communications module or it is not recognised by the EPower controller unit.
- Check that the 'IP Address', 'Subnet Mask' and 'Gateway' in the 'Comms' list are correct and unique for the network configuration in use.
- Ensure that the station name is correct between your PROFINET Master (PLC) and the EPower controller unit. In particular, if you have chosen a custom name for the EPower unit, make sure that the parameter 'PNinitName' is set to 'False' to avoid the EPower's station name being overwritten during the next boot.
- Ensure that the correct EPower GSD file has been used.
- Ensure that the network is correctly configured and the configuration has been downloaded correctly to the PROFINET Master Module.
- Ensure that the PROFINET Master Module Input and Output Parameter mapping is correctly matched. If the master is attempting to read (input) or write (output) more data than has been registered on the EPower slave, using the iTools I/O Gateway Editor, the EPower slave will refuse the connection.
- If possible, replace a faulty device with a duplicate and retest.

References

1. PI - PROFIBUS and PROFINET International www.profinet.com.
2. Siemens Automation www.automation.siemens.com.

Appendix A - Warning

Continuous Writing to Parameters

Some EPower parameters are saved in an EEPROM to ensure configuration is retained through a power cycle. Normally, these parameters do not require a periodic modification. However, if these parameters are used in the communications gateway in iTools they are continuously written to. This, in time, could lead to damage of the EEPROM device. This is indicated by an error message, 'EE Checksum Fail', which appears after the unit is power cycled. Parameters which should not be continuously written to are listed in the following tables:

Access
Access.IM
Access.Keylock
Access.ClearMemory
Access.EngineerPasscode
Access.ConfigurationPasscode
Access.QuickStartPasscode

Alarm
Alarm.AlmDis.ExternIn
Alarm.AlmLat.ExternIn
Alarm.AlmStop.ExternIn
Alarm

AnalogIP
AnalogIP.Main.Type
AnalogIP.Main.RangeHigh
AnalogIP.Main.RangeLow

AnalogOP
AnalogOP.Main.Type
AnalogOP.Main.RangeHigh
AnalogOP.Main.RangeLow
AnalogOP.AlmDis.OutputFault
AnalogOP.AlmLat.OutputFault
AnalogOP.AlmStop.OutputFault

Control
Control.Setup.NominalPV
Control.Setup.EnLimit
Control.Setup.TransferEn
Control.Setup.FFType
Control.Setup.FFGain
Control.Setup.FFOffset
Control.Setup.BleedScale
Control.Main.SP
Control.Main.TransferSpan
Control.Main.TI
Control.Limit.SP1
Control.Limit.SP2
Control.Limit.SP3
Control.Limit.TI
Control.AlmDis.ClosedLoop
Control.AlmDis.PVTransfer
Control.AlmDis.Limitation
Control.AlmLat.ClosedLoop
Control.AlmLat.PVTransfer
Control.AlmLat.Limitation
Control.AlmStop.ClosedLoop

Counter
Counter.Enable
Counter.Direction
Counter.Clock
Counter.Target

Comms
Comms.User.IP_address_2
Comms.User.IP_address_3
Comms.User.IP_address_4
Comms.User.Subnet_Mask_1
Comms.User.Subnet_Mask_2
Comms.User.Subnet_Mask_3
Comms.User.Subnet_Mask_4
Comms.User.Default_Gateway_1
Comms.User.Default_Gateway_2
Comms.User.Default_Gateway_3
Comms.User.Protocol
Comms.User.Default_Gateway_4
Comms.User.Pref_Mstr_IP_1
Comms.User.Pref_Mstr_IP_2
Comms.User.Pref_Mstr_IP_3
Comms.User.Pref_Mstr_IP_4
Comms.User.ShowMac
Comms.User.Baud
Comms.User.Address
Comms.User.Network_Version
Comms.User.Extension_Cycles
Comms.User.PninitMode
Comms.User.Parity
Comms.User.PNDevNum
Comms.User.Delay
Comms.User.UnitIdent
Comms.User.DCHP_enable
Comms.User.IP_address_1
Comms.RmtPanel.Address
Comms.RmtPanel.Baud

AnSwitch
AnSwitch.In4
AnSwitch.In5
AnSwitch.In6
AnSwitch.In7
AnSwitch.In8
AnSwitch.HighLimit
AnSwitch.LowLimit
AnSwitch.Fallback
AnSwitch.FallbackVal
AnSwitch.Select
AnSwitch.In1
AnSwitch.In2
AnSwitch.In3

Digital
Digital.Type
Digital.Invert

Energy
Energy.Type
Energy.PulseScale
Energy.PulseLen
Energy.AutoScaleUnits
Energy.UsrEnergyUnit
Energy.TotEnergyUnit

Faultdet
Faultdet.GlobalDis

FiringOP
FiringOP.LoadType
FiringOP.SafetyRamp
FiringOP.SoftStart
FiringOP.SoftStop
FiringOP.DelayedTrigger

IPMonitor
IPMonitor.In
IPMonitor.Reset
IPMonitor.Threshold
IPMonitor.AlarmDays
IPMonitor.AlarmTime

Lgc2
Lgc2.Oper
Lgc2.In1
Lgc2.In2
Lgc2.FallbackType
Lgc2.Invert
Lgc2.Hysteresis

Lgc8
Lgc8.Oper
Lgc8.In6
Lgc8.In7
Lgc8.In8
Lgc8.NumIn
Lgc8.InInvert
Lgc8.OutInvert
Lgc8.In1
Lgc8.In2
Lgc8.In3
Lgc8.In4
Lgc8.In5

Network
Network.Setup.VdipsThreshold
Network.Setup.FreqDriftThreshold
Network.Setup.ChopOffThreshold1
Network.Setup.ChopOffThreshold2
Network.Setup.ChopOffNb
Network.Setup.ChopOffWindow
Network.Setup.OverVoltThreshold
Network.Setup.UnderVoltThreshold
Network.Setup.HeatsinkPreTemp
Network.Setup.VMaximum
Network.Setup.PLFSensitivity
Network.Setup.PLUthreshold
Network.Setup.OverIThreshold
Network.Setup.HeaterType
Network.Setup.VlineNominal
Network.Setup.VloadNominal
Network.Setup.IMaximum
Network.Setup.INominal
Network.Setup.IextScale
Network.Setup.VextScale
Network.AlmDis.MissMains
Network.AlmDis.ChopOff
Network.AlmDis.PLF
Network.AlmDis.PLU
Network.AlmDis.MainsVoltFault
Network.AlmDis.PreTemp
Network.AlmDis.OverCurrent

Network (Continued)
Network.AlmDis.ThyrSC
Network.AlmDis.OpenThyr
Network.AlmDis.FuseBlown
Network.AlmDis.OverTemp
Network.AlmDis.NetworkDips
Network.AlmDis.FreqFault
Network.AlmDis.PB24VFail
Network.AlmDis.TLF
Network.AlmLat.MissMains
Network.AlmLat.PLF
Network.AlmLat.PLU
Network.AlmLat.MainsVoltFault
Network.AlmLat.PreTemp
Network.AlmLat.OverCurrent
Network.AlmLat.ThyrSC
Network.AlmLat.FuseBlown
Network.AlmLat.OverTemp
Network.AlmLat.NetworkDips
Network.AlmLat.FreqFault
Network.AlmLat.PB24VFail
Network.AlmLat.TLF
Network.AlmStop.PLF
Network.AlmStop.PLU
Network.AlmStop.MainsVoltFault
Network.AlmStop.PreTemp
Network.AlmStop.TLF

LTC
LTC.MainPrm.Type
LTC.MainPrm.TapNb
LTC.MainPrm.S1
LTC.MainPrm.S2
LTC.MainPrm.S3
LTC.AlmDis.Fuse
LTC.AlmDis.Temp
LTC.AlmLat.Fuse
LTC.AlmLat.Temp

Math2
Math2.Oper
Math2.Select
Math2.In1
Math2.In2
Math2.In1Mul
Math2.In2Mul
Math2.Units
Math2.Resolution
Math2.LowLimit
Math2.HighLimit
Math2.Fallback
Math2.FallbackVal

Modultr
Modultr.Mode
Modultr.MinOnTime
Modultr.CycleTime
Modultr.LgcMode
Modultr.SwitchPA

SetProv
SetProv.SPSelect
SetProv.SPTrack
SetProv.SPUnits
SetProv.HiRange
SetProv.RemSelect
SetProv.LocalSP
SetProv.Limit
SetProv.RampRate
SetProv.DisRamp

PLM
PLM.Main.Type
PLM.Main.Period
PLM.Station.Address
PLM.Network.Ps
PLM.AlmDis.PrOverPs
PLM.AlmLat.PrOverPs

PLMChan
PLMChan.Group
PLMChan.ShedFactor

Total
Total.In
Total.Units
Total.Resolution
Total.AlarmSP
Total.Run
Total.Hold
Total.Reset

Timer
Timer.Type
Timer.Time
Timer.In

UsrVal
UsrVal.Units
UsrVal.Resolution
UsrVal.HighLimit
UsrVal.LowLimit
UsrVal.Val
UsrVal.Status

Solution:

There are two different solutions:

1. Check the IO Gateway configuration in iTools and ensure that none of the listed parameters are present. If they are, either an alternative should be found. For example, if you want to modify the setpoint, instead of writing directly in the parameter Control.MainSP (saved in EEPROM), use a SetProv block and write into SetProv.Remote1(not saved in EEPROM). The result will be the same but it will have no effect on the life time of the EEPROM.

- In later versions of EPower (V3.01 onwards), the method of handling the IO Gateway writes is different. No parameters modified through the IO Gateway are saved in EEPROM. The saving in the EEPROM is only achieved with other methods of writing. A warning message now appears in the help of iTools which informs that the cyclic writing of these parameters is not advised.

Please contact Eurotherm for further advice.

Scaled Integers

The Modbus address is used to read/write the parameter values in a 16-bit scaled integer format. In addition, all parameters have an IEEE region Modbus address $[(2 * \text{Scaled Integer Address}) + 0x8000]$ which can be used to read/write values in native format. The 16-bit scaled integer format is widely known/used in industry, and many network masters have only the capability of reading/writing values in this format.

However, parameters which in native format have values greater than the maximum 16-bit scaled integer value (32767) have to be scaled even further to allow the values to be read/written via the 16-bit scaled integer comms address.

This section describes the mechanism to be used to scale values further when accessed via the scaled integer modbus address. In addition, it lists the parameters to be scaled in this way and what scaling will be applied to them.

Re-scaling

Parameters that require re-scaling are scaled into one of the following formats:

Kilo with 1dp (for example 124680W scaled to 124.7KW). Effective range: 100W – 3.2766MW

Kilo with 2dp (for example 124680W scaled to 124.68KW). Effective range: 10W – 327.66KW

Mega with 2dp (for example 124680W scaled to 0.12MW). Effective range: 10KW – 327.66MW

Notes:

- Scaling formats are pre-defined - they are NOT user configurable.
- Values are rounded up/down.

Parameters Which Always Require Rescaling

Some parameters within EPower will ALWAYS require re-scaling when being accessed via the scaled integer comms. These parameters will be trapped at the point of being read/written via the scaled integer comms address for the re-scaling to be applied.

Parameters which always require rescaling are listed in the following table:

Parameter	Re-scaling factor
Network.1-4.Meas.PBurst	KILO (1dp)
Network.1-4.Meas.P	KILO (1dp)
Network.1-4.Meas.S	KILO (1dp)
Network.1-4.Meas.Q	KILO (1dp)

Parameter	Re-scaling factor
Network.1-4.Meas.IsqBurst	KILO (1dp)
Network.1-4.Meas.Isq	KILO (1dp)
Network.1-4.Meas.IsqMax	KILO (1dp)
Network.1-4.Meas.VsqBurst	KILO (1dp)
Network.1-4.Meas.Vsq	KILO (1dp)
Network.1-4.Meas.VsqMax	KILO (1dp)
PLM.Network.Pmax	MEGA (2dp)
PLM.Network.Pt	MEGA (2dp)
PLM.Network.Ps	MEGA (2dp)
PLM.Network.Pr	MEGA (2dp)
PLMChan.PZmax	KILO (1dp)

Conditional Re-scaling

There are other parameters within EPower that might require re-scaling depending upon the soft wiring configuration of the instrument.

At start-up, after the wiring has been verified, the conditional scaling algorithm is called to interrogate the wiring and configure associated scaling flags appropriately for those parameters that are to be conditionally re-scaled. When these parameters are accessed via the scaled integer comms, the associated scaling flags will be tested and the appropriate scaling, if any will be applied.

The following table lists those parameters that are conditionally re-scaled and the state of the condition upon which the re-scale factor will be applied:

Parameter	Condition	Re-scaling factor
Control.1.Setup.NominalPV	When Control.1.Main.PV is wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Setup.NominalPV	When Control.2.Main.PV is wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Setup.NominalPV	When Control.3.Main.PV is wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Setup.NominalPV	When Control.4.Main.PV is wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Main.PV	When wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Main.PV	When wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Main.PV	When wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Main.PV	When wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Main.TransferPV	When wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Main.TransferPV	When wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Main.TransferPV	When wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Main.TransferPV	When wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Main.TransferSpan	When Control.1.Main.PV is wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Main.TransferSpan	When Control.2.Main.PV is wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)

Parameter	Condition	Re-scaling factor
Control.3.Main.TransferSpan	When Control.3.Main.PV is wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Main.TransferSpan	When Control.4.Main.PV is wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Limit.PV1	When wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Limit.PV1	When wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Limit.PV1	When wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Limit.PV1	When wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Limit.PV2	When wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Limit.PV2	When wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Limit.PV2	When wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Limit.PV2	When wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Limit.PV3	When wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Limit.PV3	When wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Limit.PV3	When wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Limit.PV3	When wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Limit.SP1	When Control.1.Limit.PV1 is wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Limit.SP1	When Control.2.Limit.PV1 is wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Limit.SP1	When Control.3.Limit.PV1 is wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Limit.SP1	When Control.4.Limit.PV1 is wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Limit.SP2	When Control.1.Limit.PV2 is wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Limit.SP2	When Control.2.Limit.PV2 is wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Limit.SP2	When Control.3.Limit.PV2 is wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Limit.SP2	When Control.4.Limit.PV2 is wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
Control.1.Limit.SP3	When Control.1.Limit.PV3 is wired from Network.1.Meas.P, Vsq or Isq	KILO (1dp)
Control.2.Limit.SP3	When Control.2.Limit.PV3 is wired from Network.2.Meas.P, Vsq or Isq	KILO (1dp)
Control.3.Limit.SP3	When Control.3.Limit.PV3 is wired from Network.3.Meas.P, Vsq or Isq	KILO (1dp)
Control.4.Limit.SP3	When Control.4.Limit.PV3 is wired from Network.4.Meas.P, Vsq or Isq	KILO (1dp)
SetProv.1.Remote1	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.1.WorkingSP is wired to)	KILO (1dp)
SetProv.2.Remote1	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.2.WorkingSP is wired to)	KILO (1dp)
SetProv.3.Remote1	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.3.WorkingSP is wired to)	KILO (1dp)

Parameter	Condition	Re-scaling factor
SetProv.4.Remote1	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.4.WorkingSP is wired to)	KILO (1dp)
SetProv.1.Remote2	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.1.WorkingSP is wired to)	KILO (1dp)
SetProv.2.Remote2	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.2.WorkingSP is wired to)	KILO (1dp)
SetProv.3.Remote2	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.3.WorkingSP is wired to)	KILO (1dp)
SetProv.4.Remote2	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.4.WorkingSP is wired to)	KILO (1dp)
SetProv.1.LocalSP	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.1.WorkingSP is wired to)	KILO (1dp)
SetProv.2.LocalSP	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.2.WorkingSP is wired to)	KILO (1dp)
SetProv.3.LocalSP	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.3.WorkingSP is wired to)	KILO (1dp)
SetProv.4.LocalSP	When Control.n.Main.PV is wired from Network.n.Meas.P, Vsq or Isq (where n is the instance of the control block that SetProv.4.WorkingSP is wired to)	KILO (1dp)

Energy Counter Scaling

The Energy Counters have two float32 values that have a wide dynamic range, 0 - $3e+12$ (Wh). These values already have their own scaling units that can be set to: Wh, kWh, 10kWh, 100kWh, MWh.

For the Energy Counter values to be read via the 16-bit scaled integer comms, the existing scaling units parameters require extending to include 10MWh and 100MWh thus giving a maximum value of:

$$32767 (100MWh) = 3.2767e+12 (Wh)$$

The Network Master can then read the values of the energy counter by firstly reading the units parameter, then reading the value parameter and then performing the necessary calculation.

Appendix B Communication Enhancement Modbus TCP and Modbus RTU

This appendix describes the use of the 'Anybus I/O Gateway' for Modbus TCP and Modbus RTU, to allow up to any 32 input parameters to be read as a block AND any 16 output parameters to be written as a block via user communication (Modbus TCP or Modbus RTU).

The feature allows 'block' reads and writes to be aimed at a new 'special' Modbus address.

Read / Write accesses to this 'special' Modbus address are indirected via the fieldbus I/O gateway input / output definition tables. This then allows 'block' reads and writes from and to parameters defined in the I/O gateway.

Configuring the Block Read and Write Table

EPower controller units contain many parameters, so it is necessary for the user to define which Input and Output parameters are to be available for block read and write.

The Input/Output definitions are configured using iTools. Select the 'Fieldbus I/O Gateway' tool from the lower toolbar, and an editor screen will appear similar to that shown in [Figure 64](#) below.

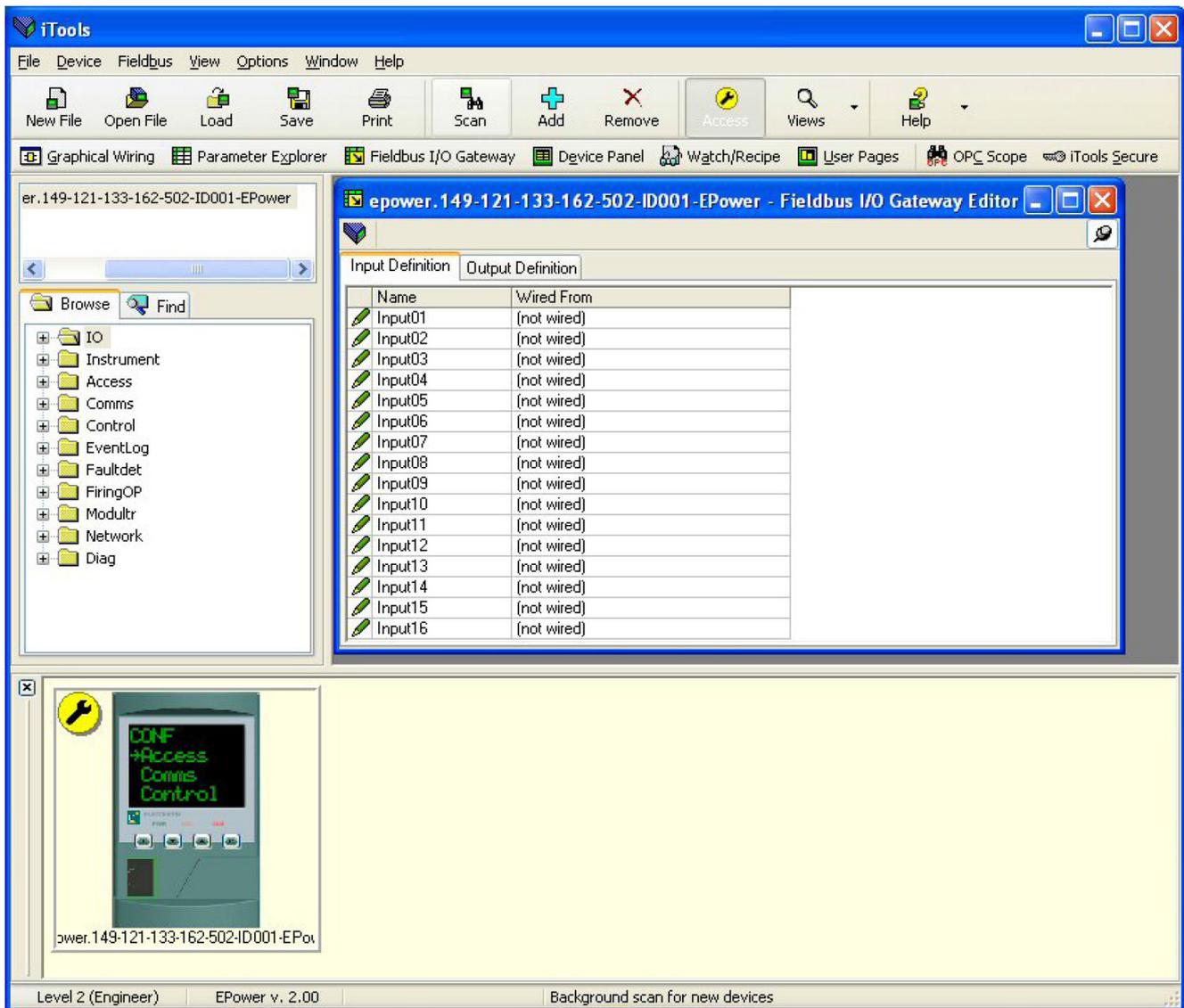


Figure 64 Fieldbus I/O Gateway Editor

There are two tabs within the editor, one for the definition of Inputs, and the other for Outputs. 'Inputs' are values sent from the EPower controller to the Modbus master, for example, alarm status information or measured values, i.e. they are readable values.

'Outputs' are values received from the master and used by the EPower controller, for example, setpoints written from the master to EPower controller.

The procedure for selecting variables is the same for both input and output tabs. Double-click the next available position in the input or output data and select the variable to assign to it. A pop-up provides a browser from which a list of parameters can be opened. Double-click the parameter to assign it to the input definition. Note that you should assign inputs and outputs contiguously, as a 'not wired' entry will terminate the list even if there are assignments following it. [Figure 65](#) shows an example of the pop-up and the input list produced.

A maximum of 32 input and 16 output parameters may be set using the Fieldbus I/O Gateway Editor.

The only way to access this table is to read at a specific address which is the first address of the table. This fixed address is 3078 (0x0C06).

In the same way and with the same fixed address, you can perform a block write to change parameters defined in output I/O gateway.

Note: With this principle, parameters defined in the I/O gateway input definition may be read by accessing 'block read' at specific address 3078. Parameters defined in the I/O gateway output definition may be written to at the same 'block write' address, 3078. Example: assume the first parameter in the I/O gateway is defined as main.PV and the first parameter in the output is defined as setpointprovider.local setpoint, then it is possible to read a value of, for example, 900 and write a value of, for example, 50.0.

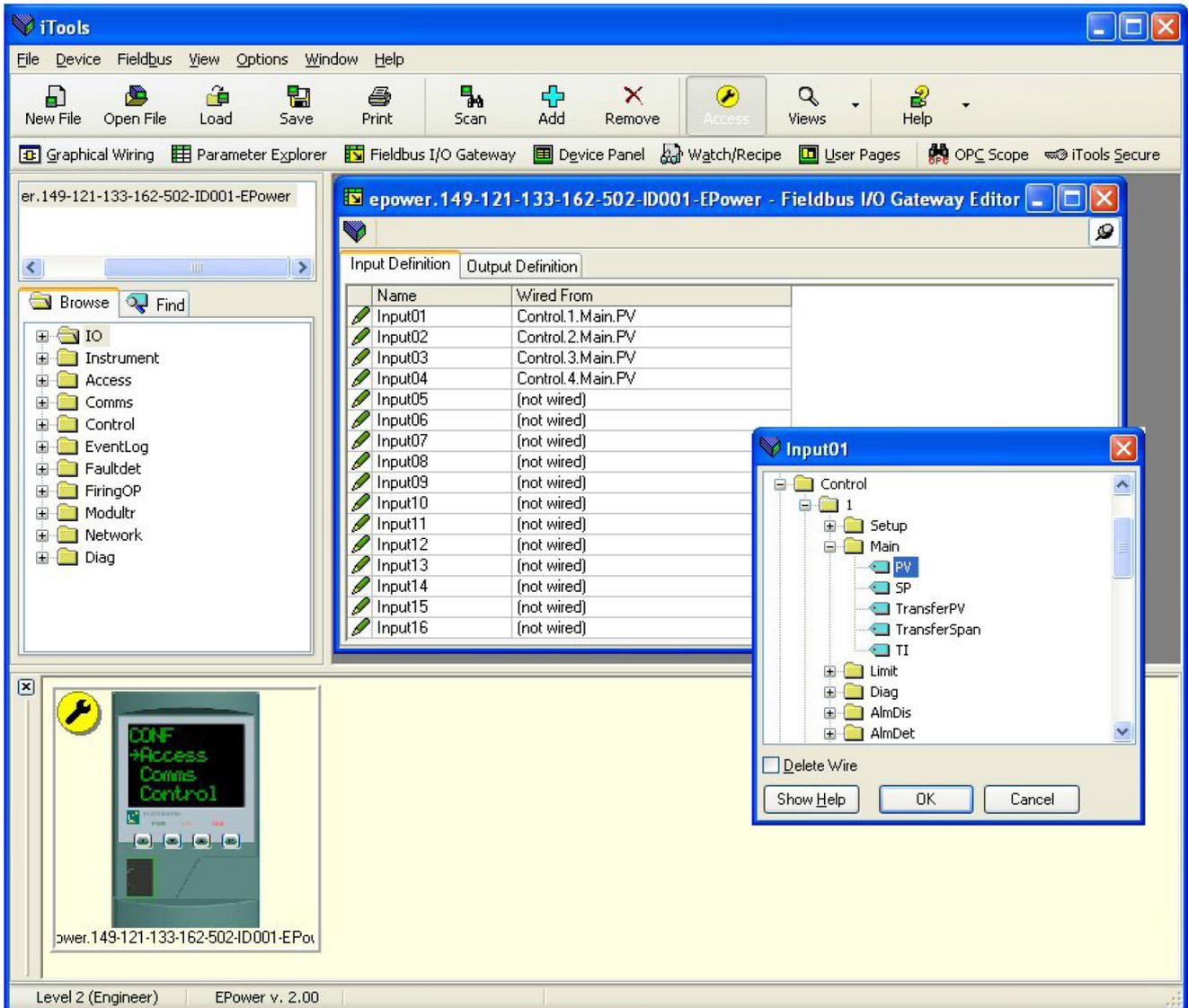


Figure 65 Assigning parameters

Appendix C. Glossary of Terms

ASCII	American Standards Committee for Information Interchange. In normal usage this refers to the character code defined by this committee for the exchange of information between devices.
Baud	The number of line signal variations per second. Used to indicate the rate at which data are transmitted on a line.
Bus	A common electrical network allowing devices, (computers, instruments) to communicate with each other.
CRC	Cyclic Redundancy Check. The CRC is an error check code and is two bytes, (16bits) long calculated from the preceding message. From a comparison of the calculated CRC and the received CRC the validity of the message can be determined.
Duplex (full duplex)	A communication channel capable of operating in both directions simultaneously.
EIA	Electrical Industries Association, the standards body that has defined electrical requirements of communications systems such as EIA232, EIA422 and EIA485 (formally RS232, RS422, RS485).
EOT	The End of Transmission segment is a period of inactivity 3.5 times the single character transmission time. The EOT segment at the end of a message indicates to the listening device that the next transmission will be a new message and therefore a device address character.
Half duplex	A communication channel capable of operating in both directions, but not simultaneously.
Message frame	A message is made up of a number of characters sequenced so that the receiving device can understand. This structure is called a message frame.
MSB	Most significant byte
LSB	Least significant byte
Non synchronous	A data channel in which no timing information is transferred between communicating devices.
Parity	A mechanism used for the detection of transmission errors when single characters are being transmitted. A single binary digit known as the parity bit has a value of 0 or 1 depending on the number of '1's in a data message. This allows single bit error detection in the receiver.
RTU	Remote Terminal Unit. This refers to the code used for the exchange of information between devices.
RX	Receiver on a communication bus.
Simplex	A communication channel capable of operating in one direction only.
Start bit	A voltage level used to signal the start of a character transmission frame
Stop bit	A voltage level used to signal the end of a character transmission frame
TX	Transmitter on a communication bus

Appendix D. ASCII Codes

HEX-ASCII TABLE					
00	NUL	2B	+	56	V
01	SOH	2C	,	57	W
02	STX	2D	-	58	X
03	ETX	2E	.	59	Y
04	EOT	2F	/	5A	Z
05	ENQ	30	0	5B	[
06	ACK	31	1	5C	\
07	BEL	32	2	5D]
08	BS	33	3	5E	^
09	HT	34	4	5F	-
0A	LF	35	5	60	`
0B	VT	36	6	61	a
0C	FF	37	7	62	b
0D	CR	38	8	63	c
0E	SO	39	9	64	d
0F	SI	3A	:	65	e
10	DLE	3B	;	66	f
11	DC1(X-ON)	3C	<	67	g
12	DC2	3D	=	68	h
13	DC3(X-OFF)	3E	>	69	l
14	DC4	3F	?	6A	j
15	NAK	40	@	6B	k
16	SYN	41	A	6C	l
17	ETB	42	B	6D	m
18	CAN	43	C	6E	n
19	EM	44	D	6F	o
1A	SUB	45	E	70	p
1B	ESC	46	F	71	q
1C	FS	47	G	72	r
1D	GS	48	H	73	s
1E	RS	49	I	74	t
1F	US	4A	J	75	u
20	space	AB	K	76	v
21	!	4C	L	77	w
22	“	4D	M	78	x
23	£	4E	N	79	y
24	\$	4F	O	7A	z
25	%	50	P	7B	{
26	&	51	Q	7C	
27	'	52	R	7D	}
28	(53	S	7E	~

HEX-ASCII TABLE					
29)	54	T	7F	DEL
2A	*	55	U		



Scan for local contents

Eurotherm Ltd

Faraday Close
Durrington
Worthing
West Sussex
BN13 3PL
Phone: +44 (0) 1903 268500
www.eurotherm.com

Eurotherm by Schneider Electric, the Eurotherm logo, Chessell, EurothermSuite, Mini8, Eycon, Eyris, EPower, EPack, nanodac, piccolo, versadac, optivis, Foxboro and Wonderware are trademarks of Schneider Electric, its subsidiaries and affiliates. All other brands may be trademarks of their respective owners.

All rights are strictly reserved. No part of this document may be reproduced, modified or transmitted in any form by any means, neither may it be stored in a retrieval system other than for the purpose to act as an aid in operating the equipment to which the document relates, without the prior written permission of Eurotherm Limited.

Eurotherm Limited pursues a policy of continuous development and product improvement. The specifications in this document may therefore be changed without notice. The information in this document is given in good faith, but is intended for guidance only. Eurotherm Limited will accept no responsibility for any losses arising from errors in this document.

© 2020 Eurotherm Limited. All rights reserved.

HA179770 issue 8 CN37710