

EPower™ Controller

Power management and control units

Communications Manual

DIGITAL COMMUNICATIONS MANUAL

LIST OF CONTENTS

SECTION	PAGE
1. CHAPTER 1 INTRODUCTION	3
1.1 JBUS v MODBUS	3
1.2 References	3
1.3 EIA232, EIA422 and EIA485 Transmission Standards	4
1.4 Cable Selection	5
1.5 Wiring General	6
1.6 Precautions	6
1.7 Grounding	6
2. CHAPTER 2 DIGITAL COMMUNICATIONS HARDWARE	7
2.1 Communications pinouts	7
2.1.1 Modbus RTU.....	7
2.1.2 Modbus TCP (Ethernet 10baseT).....	7
2.1.3 DeviceNet.....	8
2.1.4 Profibus	8
2.2 Wiring 2-wire EIA485	9
2.2.1 Interconnection Diagram 3-Wire EIA485	10
FIGURE 2-6: RS485 THREE-WIRE INTERCONNECTIONS (EPOWER CONTROLLER UNITS ONLY)	10
2.3 iTools	11
2.4 modbus parameters	12
2.4.1 Comms Identity	12
2.4.2 Protocol.....	12
2.4.3 Unit Address.....	12
2.4.4 Baud Rate.....	12
2.4.5 Parity	12
2.4.6 Delay.....	12
3. CHAPTER 3 MODBUS AND JBUS PROTOCOL	13
3.1 Protocol Basics.....	13
3.2 Typical Transmission Line Activity	14
3.3 Device Address.....	14
3.4 Parameter Address.....	14
3.5 Parameter Resolution.....	14
3.6 Mode of Transmission	15
3.7 Message Frame Format	15
3.8 Cyclic Redundancy Check	16
3.9 Example of a CRC Calculation	17
3.10 Example of a CRC Calculation in the 'C' Language.....	19
3.11 Example of a CRC Calculation in BASIC Language.....	20
3.12 Function Codes	21
3.13 Read n Words.....	22
3.14 Write a Word	23
3.15 Diagnostic Loopback	24
3.16 Write n Words.....	25
3.17 Error Response	26
3.18 Wait Period	27
3.19 Latency	27
3.20 Message Transmission Time	27
3.21 Status words	28
3.22 Fieldbus Comm Status.....	28
3.23 Strategy Status Word.....	29
3.24 Error Status Words.....	29
3.24.1 Status word to indicate instrument errors via comms	29

3.25	configuration mode parameters	29
4.	CHAPTER 4 MODBUS ADVANCED TOPICS	30
4.1	Access to Full Resolution Floating Point and Timing Data.....	30
4.2	Data types used in EPower controller units.....	31
4.3	Enumerated, Status Word, and Integer parameters.....	31
4.4	Floating Point Parameters	32
4.5	Time Type Parameters	33
4.6	Programmable Logic Controllers and EPower controller	33
5.	CHAPTER 5 ETHERNET	34
5.1	Overview.....	34
5.1.1	Support for other Ethernet utilities	34
5.2	Ethernet Wiring	34
5.3	Instrument setup	35
5.3.1	Unit Identity.....	35
5.3.2	Dynamic Host Configuration Protocol (DHCP) Settings.....	35
5.3.3	Default Gateway	35
5.3.4	Preferred Master	35
5.4	iTools Setup.....	36
6.	CHAPTER 6 PROFIBUS.....	37
6.1	Introduction	37
6.2	Wiring general	38
6.3	cable selection.....	38
6.4	Maximum baud rate compared with cable length	38
6.5	node address	39
6.6	Configuring the Data exchange	40
6.7	To Configure the Profibus Master.....	42
6.8	DPV1 Acyclic Communications	42
6.9	trouble-shooting.....	42
7.	CHAPTER 6 DEVICENET	43
7.1	INTRODUCTION	43
7.1.1	EPower Controller DeviceNet Features.....	43
7.2	DeviceNet Wiring.....	44
7.3	Setting up the EPower controller unit	45
7.3.1	Unit Address.....	45
7.3.2	Baud Rate.....	45
7.4	data exchange mapping.....	45
7.5	Configuring the Data exchange	46
7.6	Setting up the master	48
7.7	Establishing Communications	48
7.8	data formats.....	48
7.9	explicit messaging	48
7.10	THE EDS FILE.....	48
7.11	TROUBLESHOOTING.....	49
8.	APPENDIX A. GLOSSARY OF TERMS	50
9.	APPENDIX B. ASCII CODES.....	51
10.	INDEX	52

Issue Status of this Manual

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

1. CHAPTER 1 INTRODUCTION

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

Chapter 1 provides general information on standards, cabling and electrical connections.

Chapter 2 gives connector and wiring details for a serial link

Chapter 3 describes Modbus RTU and Jbus protocols.

Chapter 4 covers advanced topics such as access to full resolution floating point data and user interface permissions.

Chapter 5 describes Ethernet (Modbus TCP)

Chapter 6 describes Profibus

Chapter 7 describes DeviceNet®

Appendix A is a Glossary of Terms.

Appendix B lists ASCII codes.

☺ This symbol indicates a helpful hint whenever it appears.

1.1 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.

1.2 REFERENCES

Refer to the documents 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.com
	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.	

1.3 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 ± 12 volts 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:-

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**.

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 ± 200 mV 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.

The 3-wire EIA485 should be used where possible for new installations where multi-drop capability is required.

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)

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.

Most PC's 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 section 2.2.1 for wiring information for this unit, or refer to KD485 Installation and Operation Handbook.

1.4 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
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 *

1.5 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.

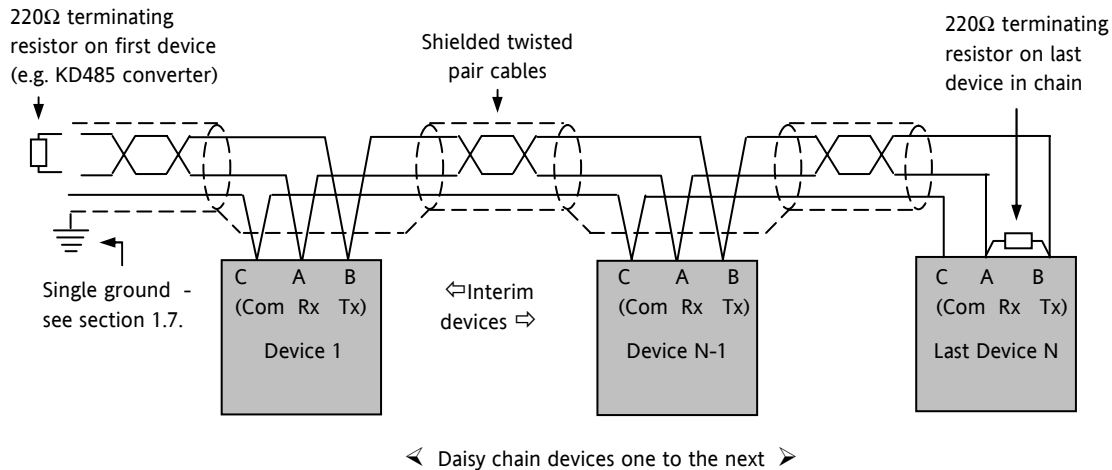


Figure 1-1: EIA485 3-wire Transmission

1.6 PRECAUTIONS

EPower controller units have fully isolated communications circuitry. In most PCs the comms ports are also fully isolated. In conditions where excessive static build up may be experienced there is a danger that damage could occur to PC or controllers. In these circumstances it may be an advantage to add a high value resistor (1MΩ for example) between the common connection and ground. Under normal circumstances this resistor is not necessary.

☺ EPower controller units are interconnected using RJ45 connectors. Suitable cables are available from Eurotherm as detailed in section 2.2.

1.7 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.

2. CHAPTER 2 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:

2.1 COMMUNICATIONS PINOUTS

2.1.1 Modbus RTU

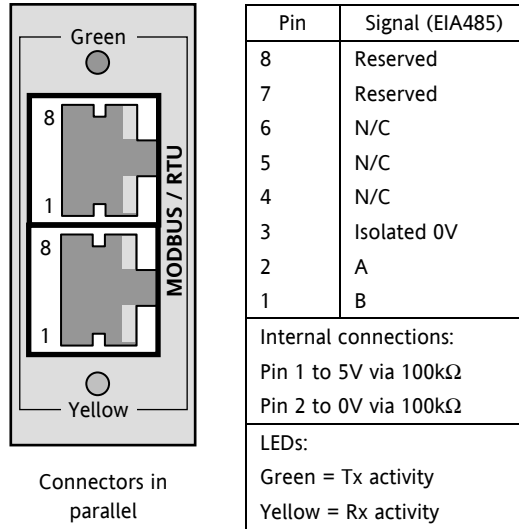


Figure 2-1: Modbus RTU pinout

2.1.2 Modbus TCP (Ethernet 10baseT)

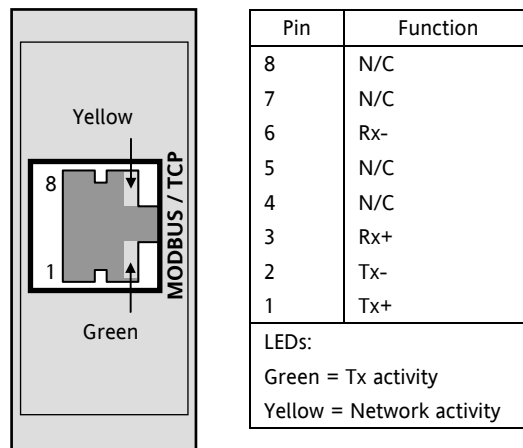


Figure 2-2: Modbus TCP (Ethernet 10baseT) pinout

2.1.3 DeviceNet

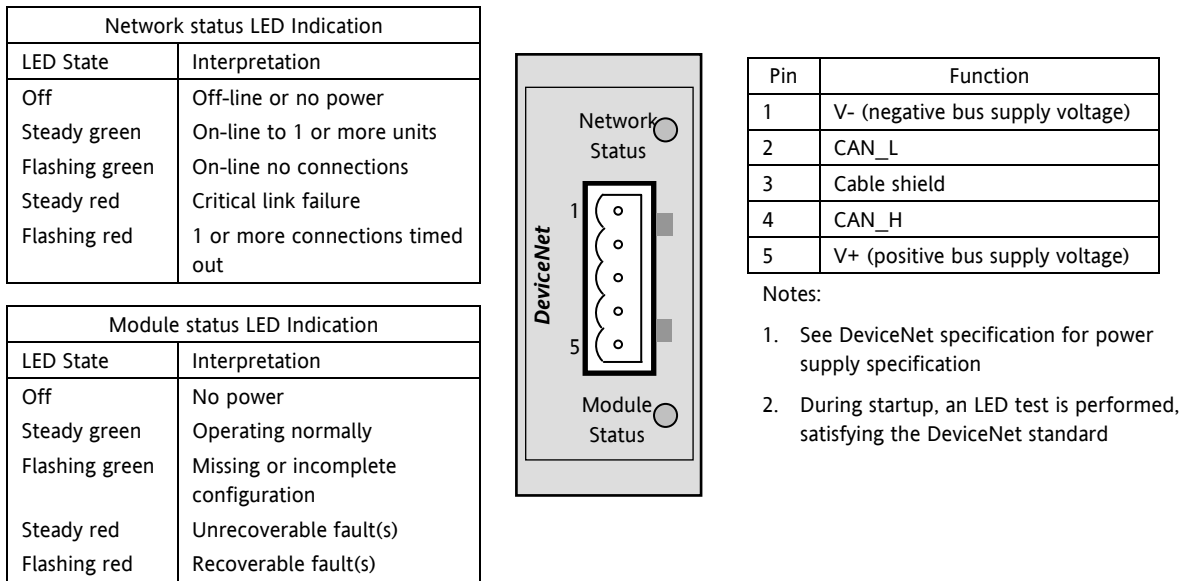


Figure 2-3: DeviceNet connector pinout

2.1.4 Profibus

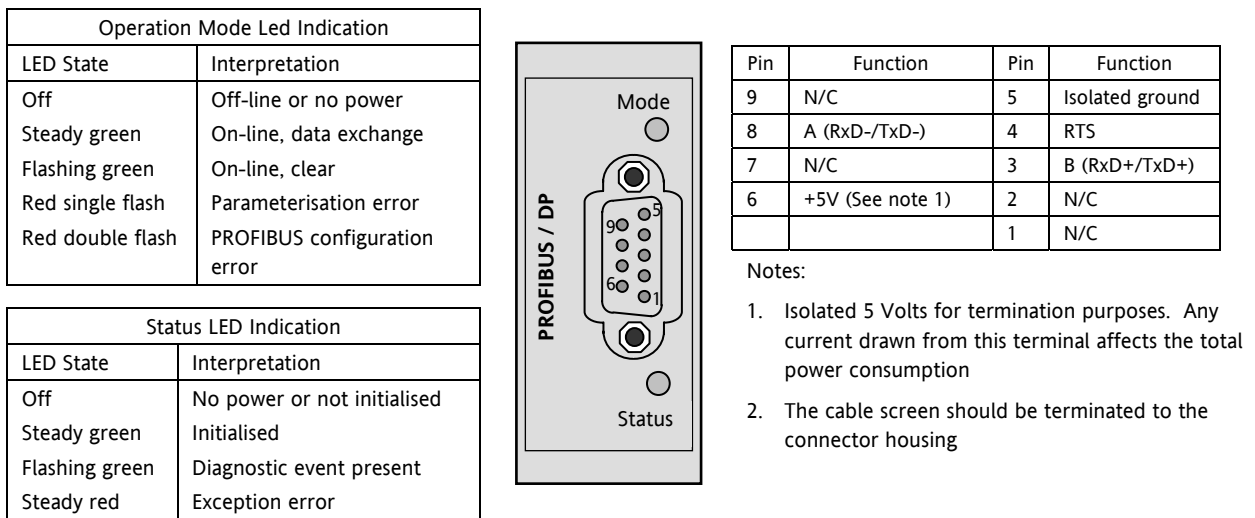


Figure 2-4: Profibus connector pinout

2.2 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 section 2.1.1).

(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

(2) All network communications lines must be terminated using the appropriate impedance as shown in Figure 1-1. 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:-

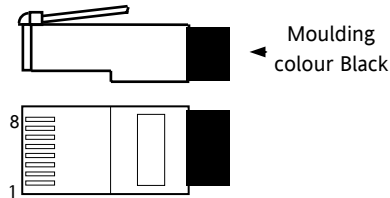


Figure 2-5: Line termination for Modbus

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 section 1.4, crimp an RJ45 plug on the EPower controller end.

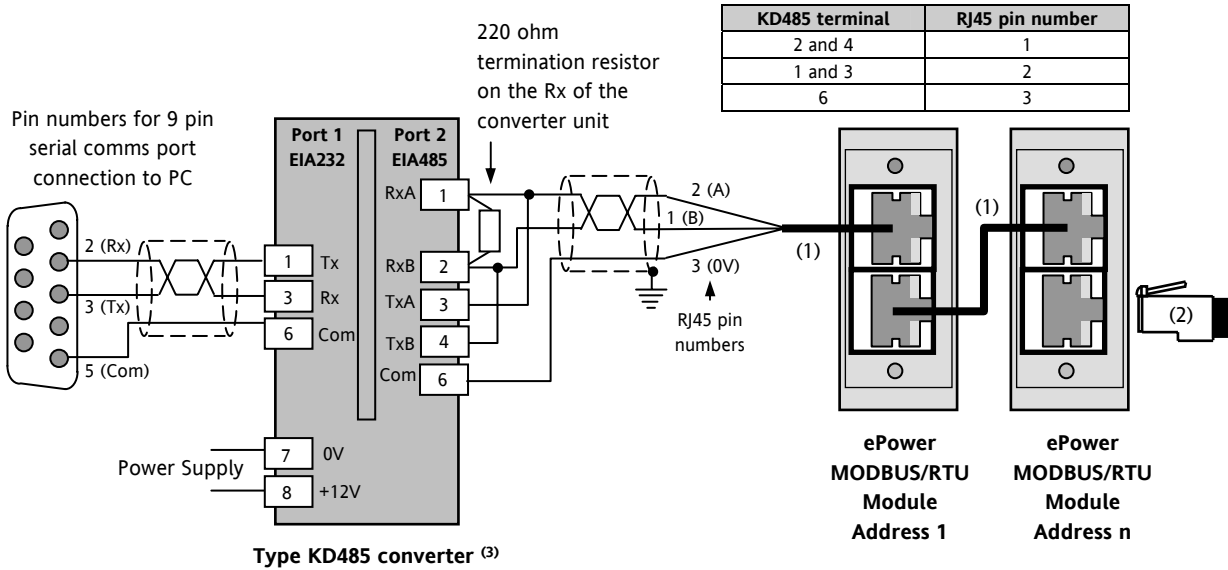
The PC serial port may be either 9 pin (most common) or 25 pin. The table below lists the correct connections for either of these ports.

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		

2.2.1 Interconnection Diagram 3-Wire EIA485

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

It may be necessary to cut a standard RJ45 cable in order to connect it to the KD485 terminals. In this case connect the wires as shown.



- (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 2-6: RS485 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 and correct line terminations (Figure 1-1) should be observed.

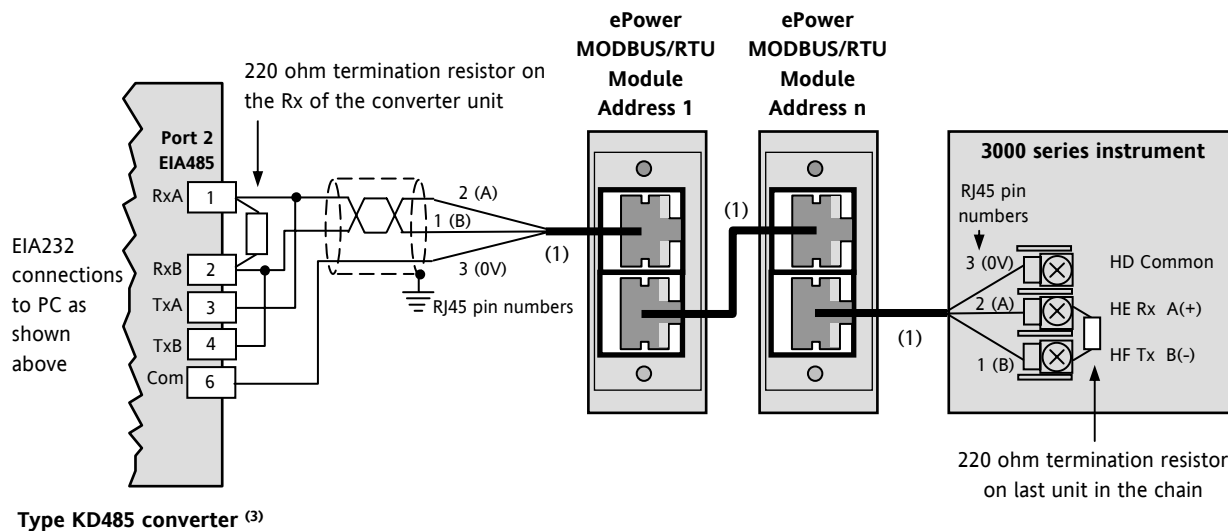



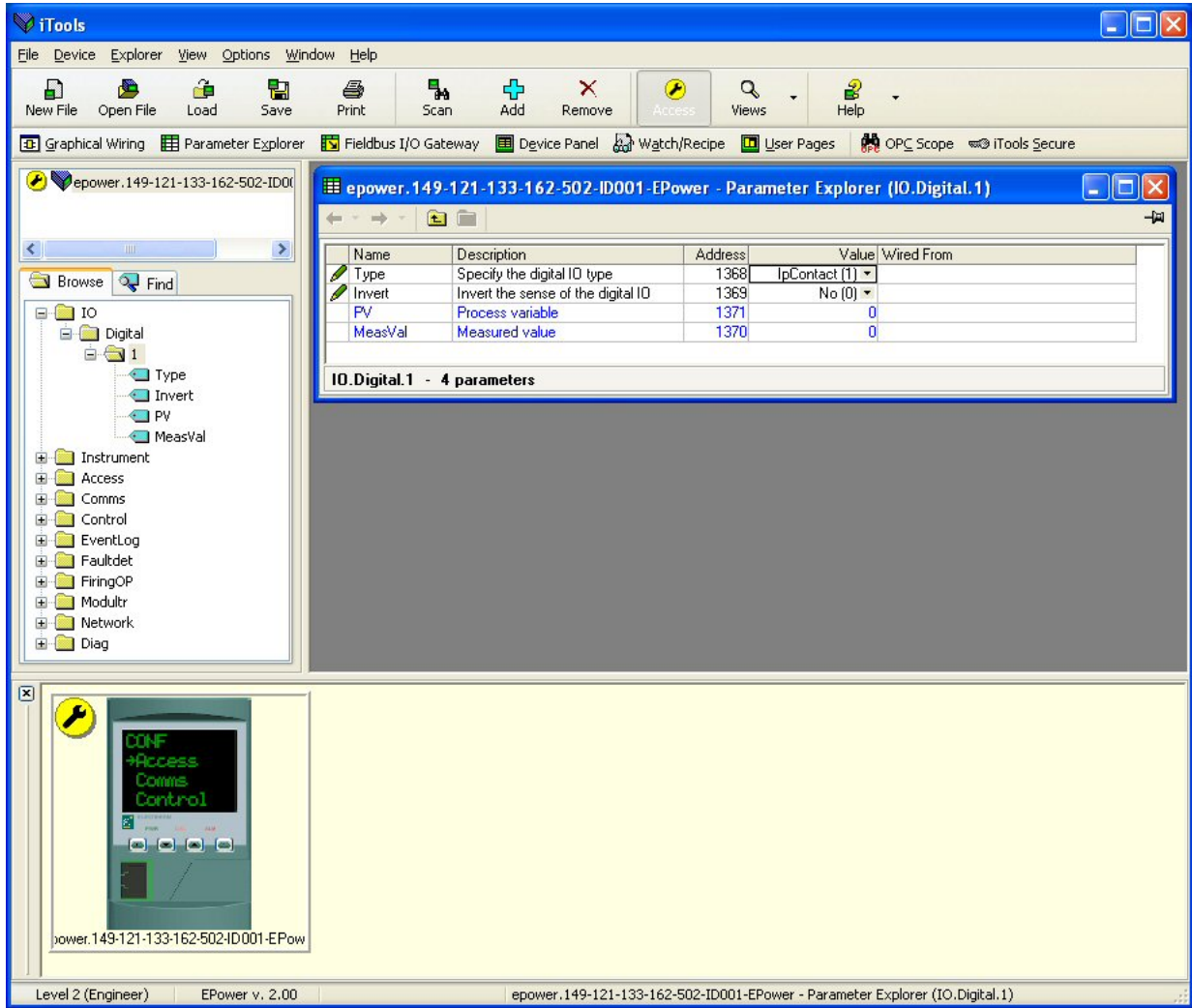
Figure 2-7: EIA485 3-wire (EPower controller and Series 3000 units)

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.

2.3 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.co.uk.

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.co.uk.

2.4 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.

2.4.1 Comms Identity

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

2.4.2 Protocol

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

2.4.3 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.

2.4.4 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 section 2.2.

Although the baud rate is an important factor, when calculating the speed of communications in a system it is often the 'latency' (section 3.19) 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.

2.4.5 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 section 3.8) 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.

2.4.6 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.

3. CHAPTER 3 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.

3.1 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;

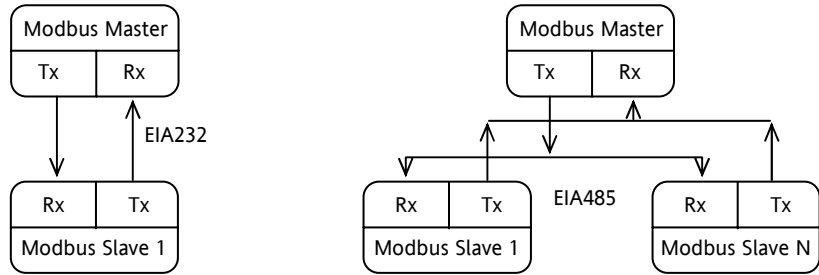


Figure 3-1: Single Serial Link

Multi Drop Serial Link

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

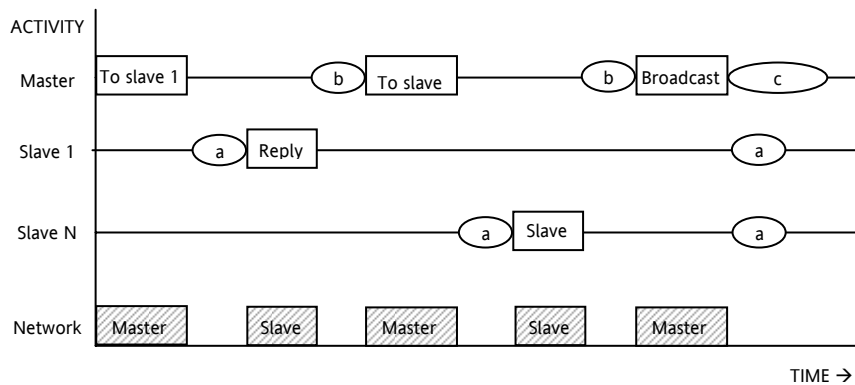
The message in either direction will consist of the following information;

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.

3.2 TYPICAL TRANSMISSION LINE ACTIVITY

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



- Period 'a' The processing time, (latency), required by the slave to complete the command and construct a reply.
- Period 'b' The processing time required by the master to analyse the slave response and formulate the next command.
- Period 'c' The wait time calculated by the master for the slaves to perform the operation. None of the slaves will reply to a broadcast message.

Figure 3-2: Modbus Timing Diagram

For a definition of the time periods required by the network, refer to 'Wait Period' in section 3.17 'Error Response'.

3.3 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.

3.4 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.

3.5 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 Chapter 4 of this manual.

3.6 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;

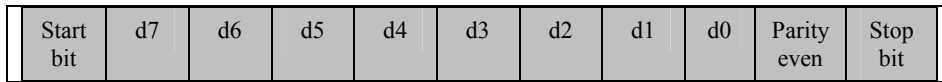
A start bit, eight data bits, a parity bit and one or two stop bits

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:



3.7 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.

3.8 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.
- 4a 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.

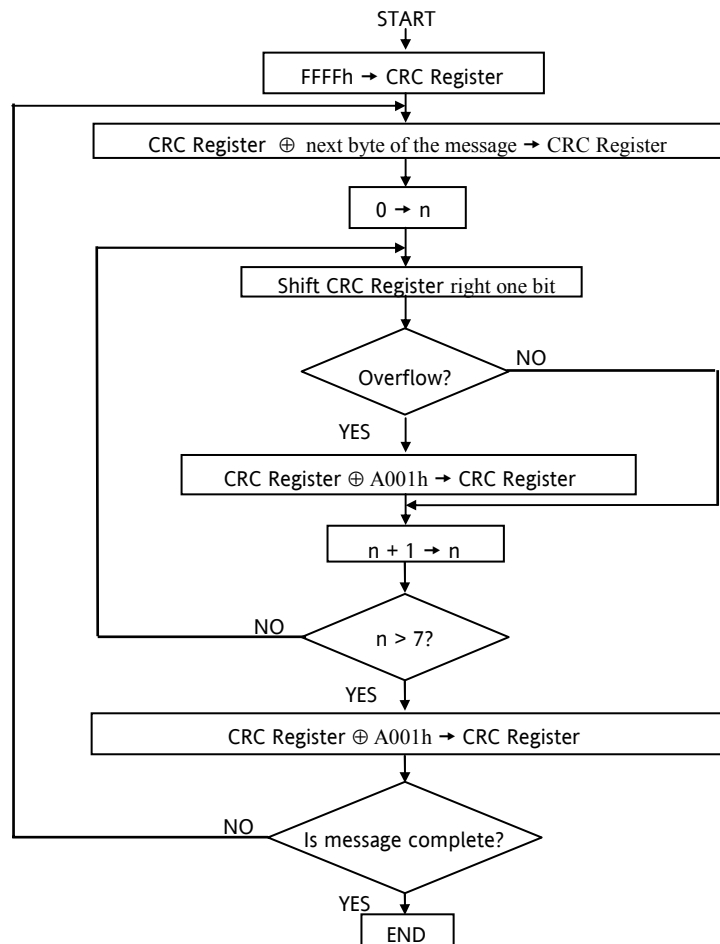


Figure 3-3: CRC Flow Chart

3.9 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	
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 ↑	

3.10 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;
}
```

3.11 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
```

3.12 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.

3.13 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.

3.14 WRITE A WORD

Function code: 06, (06h)

Command:

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

Reply:

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

The reply to function 06 is the same as the command. See section 3.17 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

3.15 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	Diagnostic Code		Loopback Data		CRC	
	08	0000					
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	Diagnostic Code		Loopback Data		CRC	
	08	0000					
02	08	00	00	12	34	ED	4F

Reply:

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

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

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	
		MSB	LSB	MSB	LSB	MSB	LSB
	10						
1 byte	1 byte						

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	
		MSB	LSB	MSB	LSB			MSB	LSB
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	
		MSB	LSB	MSB	LSB	MSB	LSB
02	10	03	9B	00	03	F1	90

3.17 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

3.18 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.

3.19 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.

3.20 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, section 3.6).

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).

3.21 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:

3.22 FIELDBUS COMM 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

3.23 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

3.24 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.

3.24.1 Status word to indicate instrument errors via comms

The bits in the status word indicate the following errors:

Bit No.	Description
0	Atleast 1 Fatal error has occurred
1	Atleast 1 Config error has occurred
2	Atleast 1 HW MissMatch error has occurred
3	Atleast 1 power module ribbon cable error, eeprom error, revision error has occurred
4	Atleast 1 Power module is running using default calibration
5	Atleast 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

3.25 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.

4. CHAPTER 4 MODBUS ADVANCED TOPICS

4.1 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 10th of seconds or 10th 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.

4.2 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 section 3.21.
- 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.

4.3 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.

4.5 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

4.6 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.

5. CHAPTER 5 ETHERNET

5.1 OVERVIEW

EPower controller units support the Modbus/TCP protocol using Ethernet. This protocol embeds the standard Modbus protocol within an Ethernet TCP layer.

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.

5.1.1 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.

5.2 ETHERNET WIRING

Ethernet capability is provided by an interface board installed within the unit and provides a single RJ45 socket (section 2.1.2). 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 meters. 10baseT operates at 10 Mbps and uses baseband transmission methods.

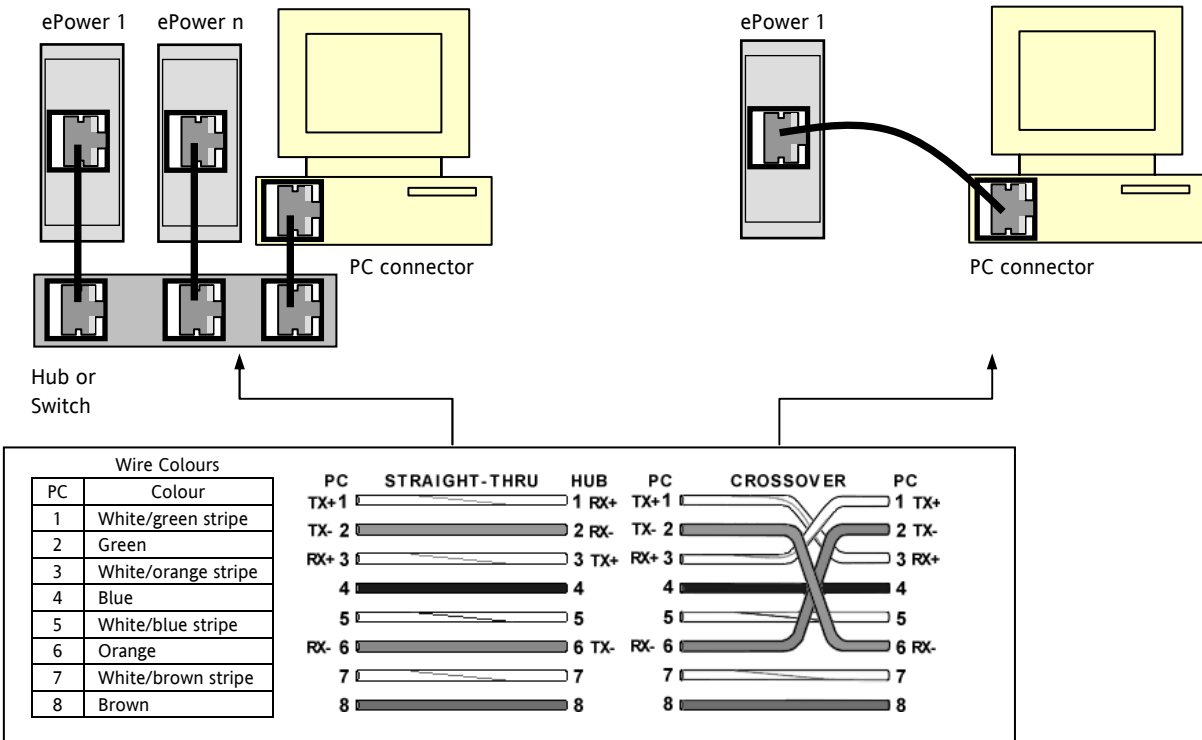


Figure 5-1: Ethernet Wiring - Single and Multiple EPower controller Units

5.3 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.

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 IPAddr1 = abc, IPAddr2 = def, IPAddr3 = ghi and IPAddr4 = 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.

5.3.1 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.

5.3.2 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.

5.3.2.1 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.

5.3.2.2 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.

5.3.3 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.

5.3.4 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.

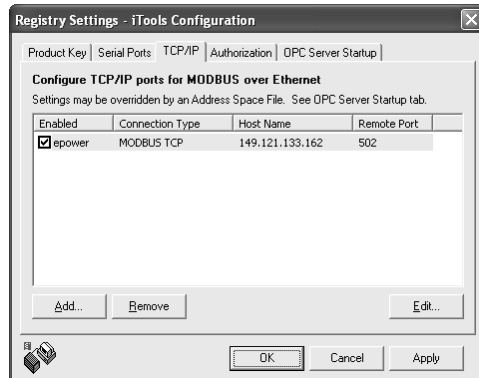
5.4 ITOOLS SETUP

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

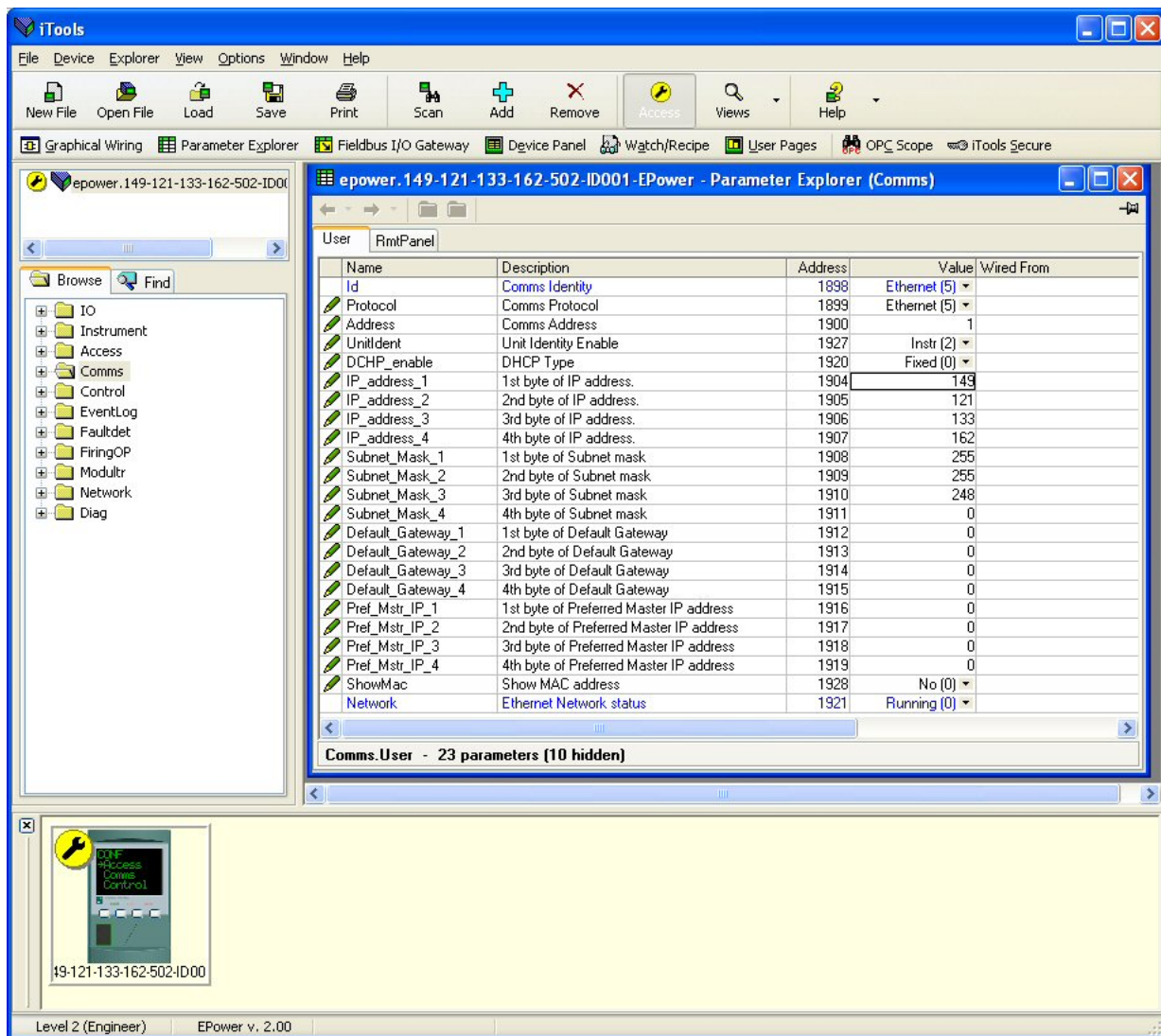


Figure 5-2: Ethernet Communications Parameters

6. CHAPTER 6 PROFIBUS

6.1 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 section 1.5), 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 – section 2.2.1, (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 2.1.4.
- 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.

6.2 WIRING GENERAL

The wiring principles described in section 1.5 apply to Profibus. Line termination is different from standard EIA485 as shown in Figure 6-1.

The Profibus cable (section 6.3) 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.

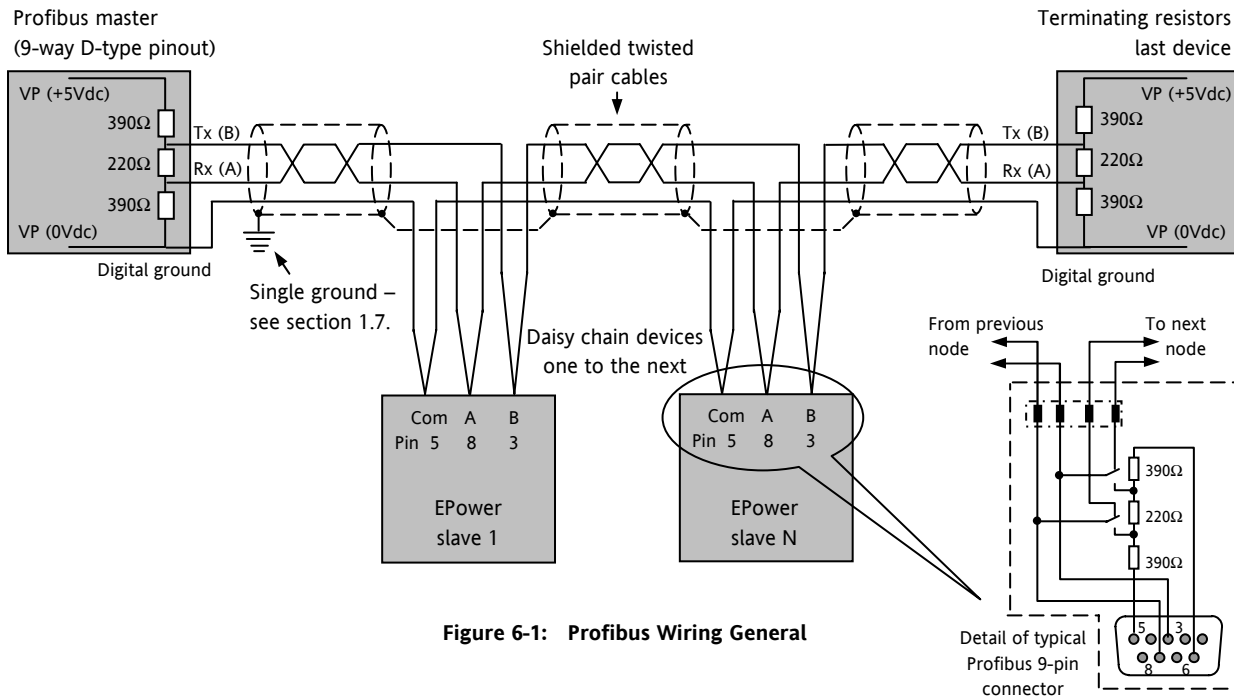


Figure 6-1: Profibus Wiring General

6.3 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

6.4 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. Guaranteed 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

6.5 NODE ADDRESS

Each node must be given a unique address. This may 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.

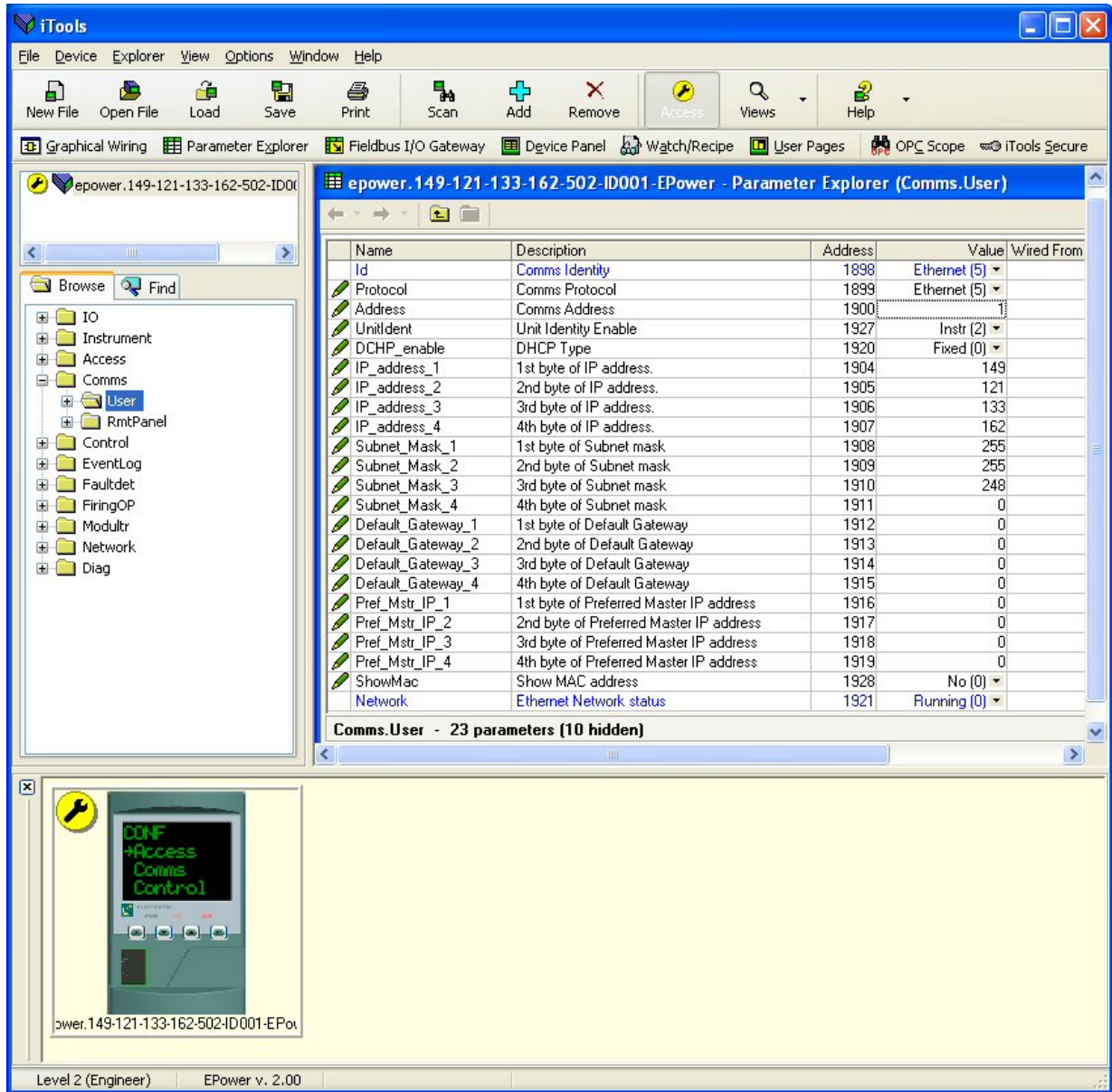


Figure 6-2: Setting the Profibus Comms Address in iTools

6.6 CONFIGURING THE DATA EXCHANGE

The Profibus 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 Profibus 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, (such as a ladder logic 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 Profibus are configured using iTools. Select the 'Fieldbus I/O Gateway' tool from the lower toolbar, and an editor screen will appear similar to the picture below.

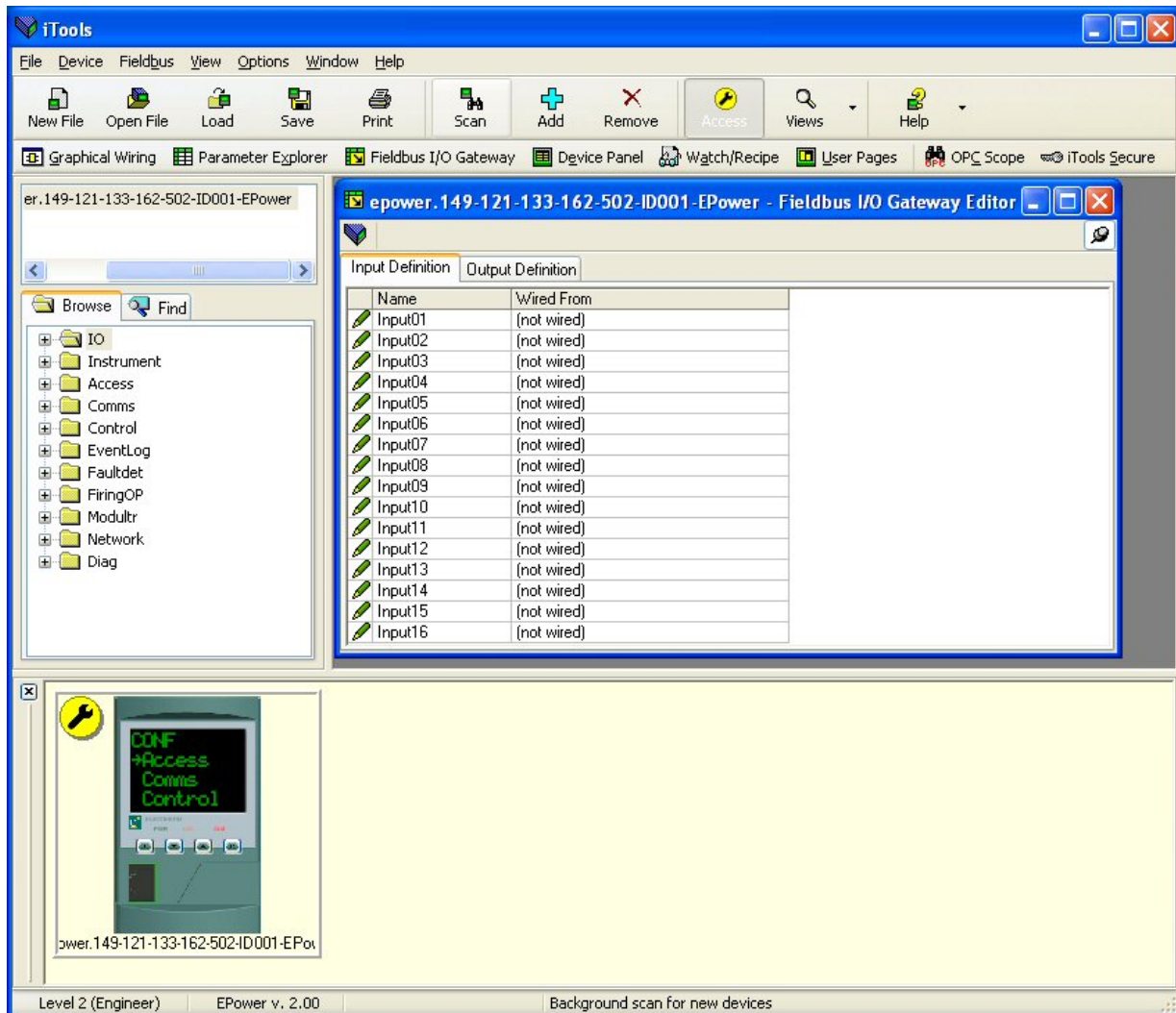


Figure 6-3: The Fieldbus I/O Gateway Editor in iTools

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.

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 6-4 shows an example of the pop-up and the input list produced.

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

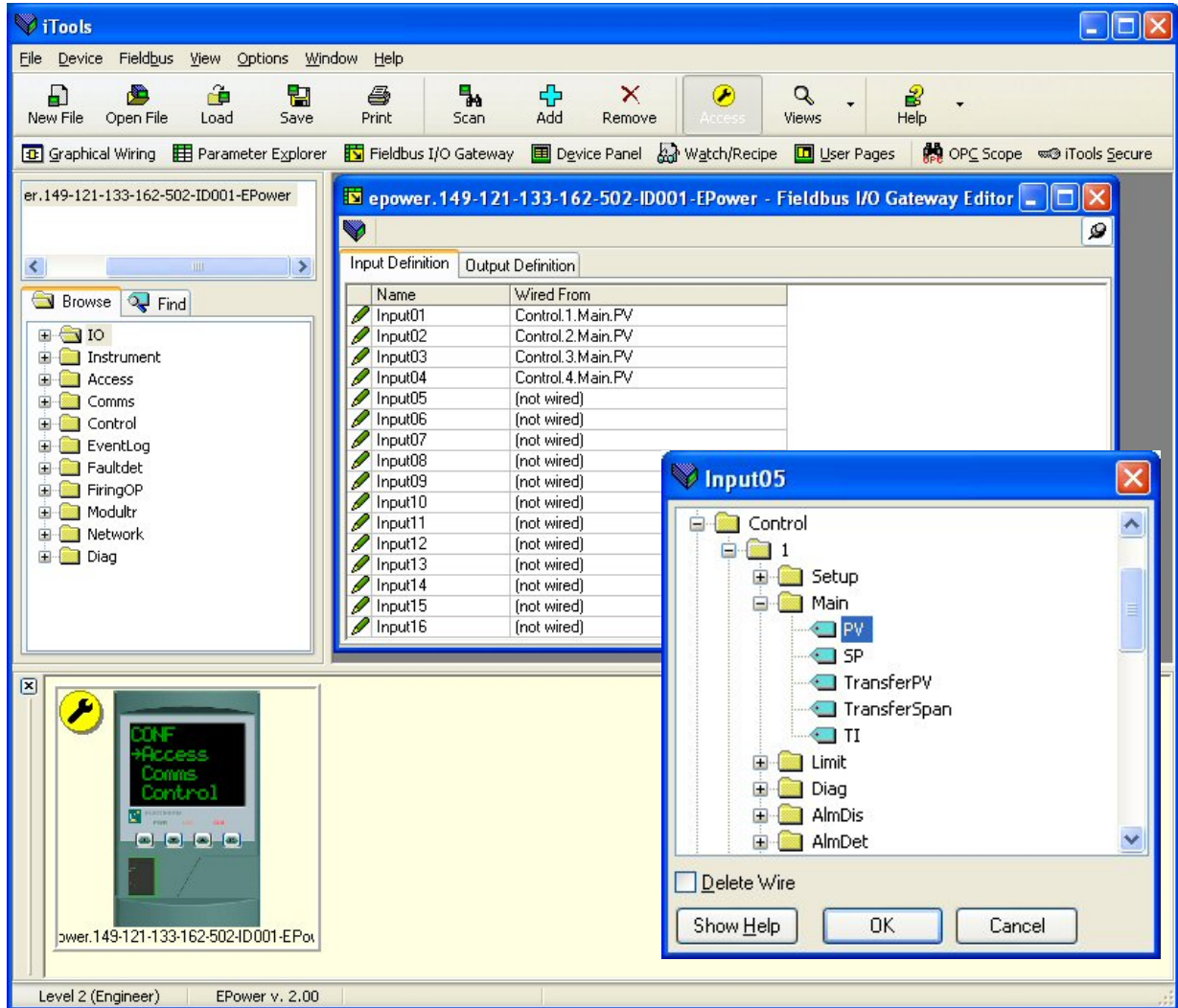



Figure 6-4: Selecting an Input Value and Example of an Input List

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 Profibus Master. In the example above, there are four input values.

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 Profibus cyclic communications 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 Profibus master.

6.7 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. The GSD file is a text file in a predefined standardised format. It is used to declare the new slave to the master and to allow communications between the master and the slave. A GSD file for EPower controller units is available on the support disc or may be downloaded from www.eurotherm.com.

Only general indications on how this is done can be provided, since this is Profibus Master specific, however the 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 available, "Input 1 word", and "Output 1 word". You must assign one module per item in the input and output lists, with the input lists being assigned first. So for the example given in the previous section, you should add the "Input 1 word" four 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.
4. 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 showing that communication is established.
5. If the light flashes, does not turn on, or lights as red, verify the proceeding steps have been correctly performed.

6.8 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:

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

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

6.9 TROUBLE-SHOOTING

No Communications

1. Check the wiring carefully, verifying the continuity of the A and B connections to the master. Ensure correct terminals have been used.
2. Check the node address is correct for the network configuration in use. Ensure the address is unique.
3. Ensure that the network has been correctly configured and that the configuration has been correctly downloaded to the master.
4. 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 6-1, 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 6-1, and that only the first and final nodes are so terminated.

7. CHAPTER 7 DEVICENET

7.1 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, see section 2.1.3. 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 section 7.5.

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	Section 7.2
Setting up EPower controller units	Section 7.3
Setting up the master using EDS files	Section 7.6
Establishing communications	Section 7.7
Transferring data on the network	

7.1.1 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

7.2 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 7-1 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)

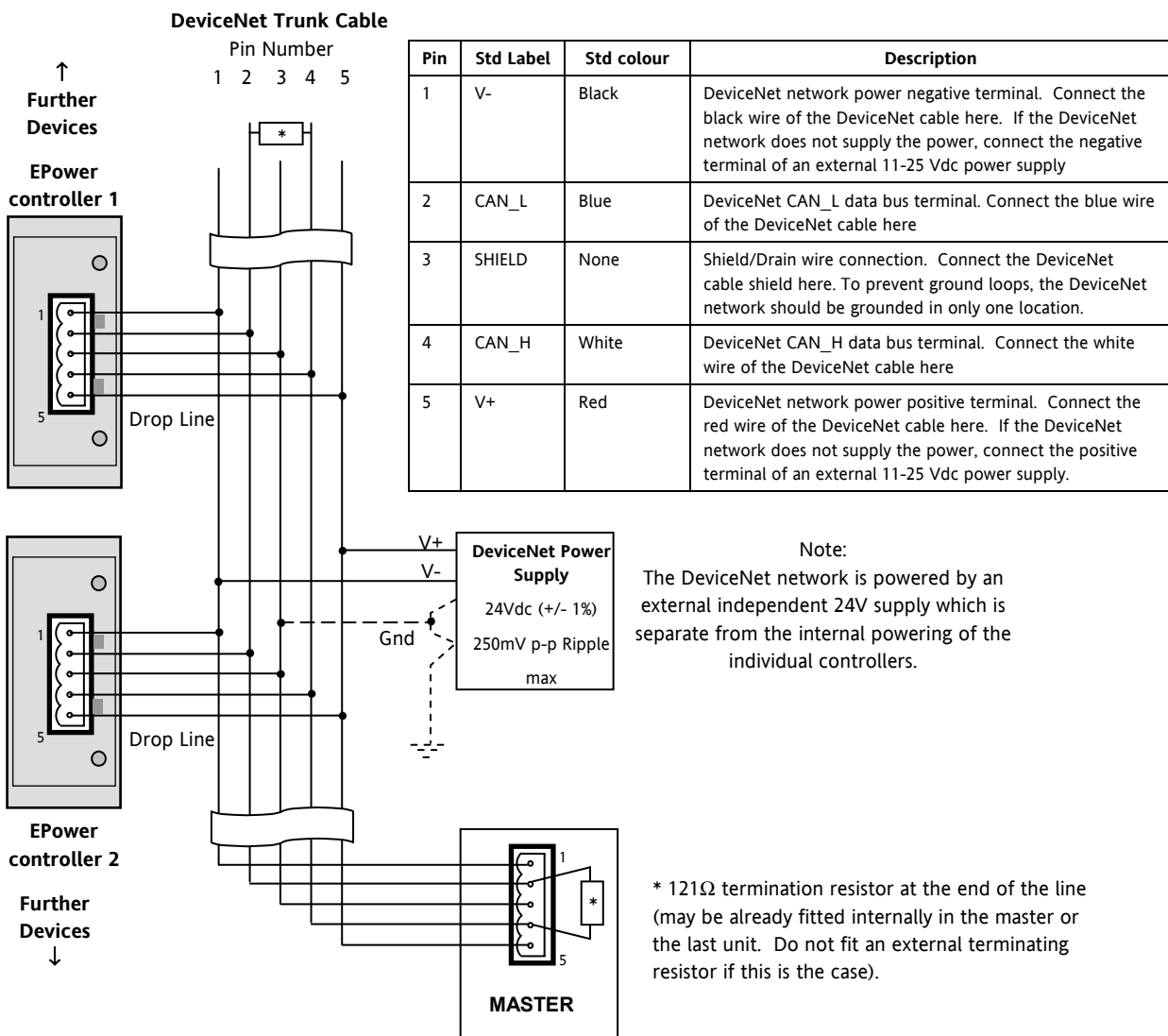


Figure 7-1: Example of DeviceNet Installation

7.3 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.

7.3.1 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. 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.

7.3.2 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.

7.4 DATA EXCHANGE MAPPING

Up to 16 input and 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	Input 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.

7.5 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.

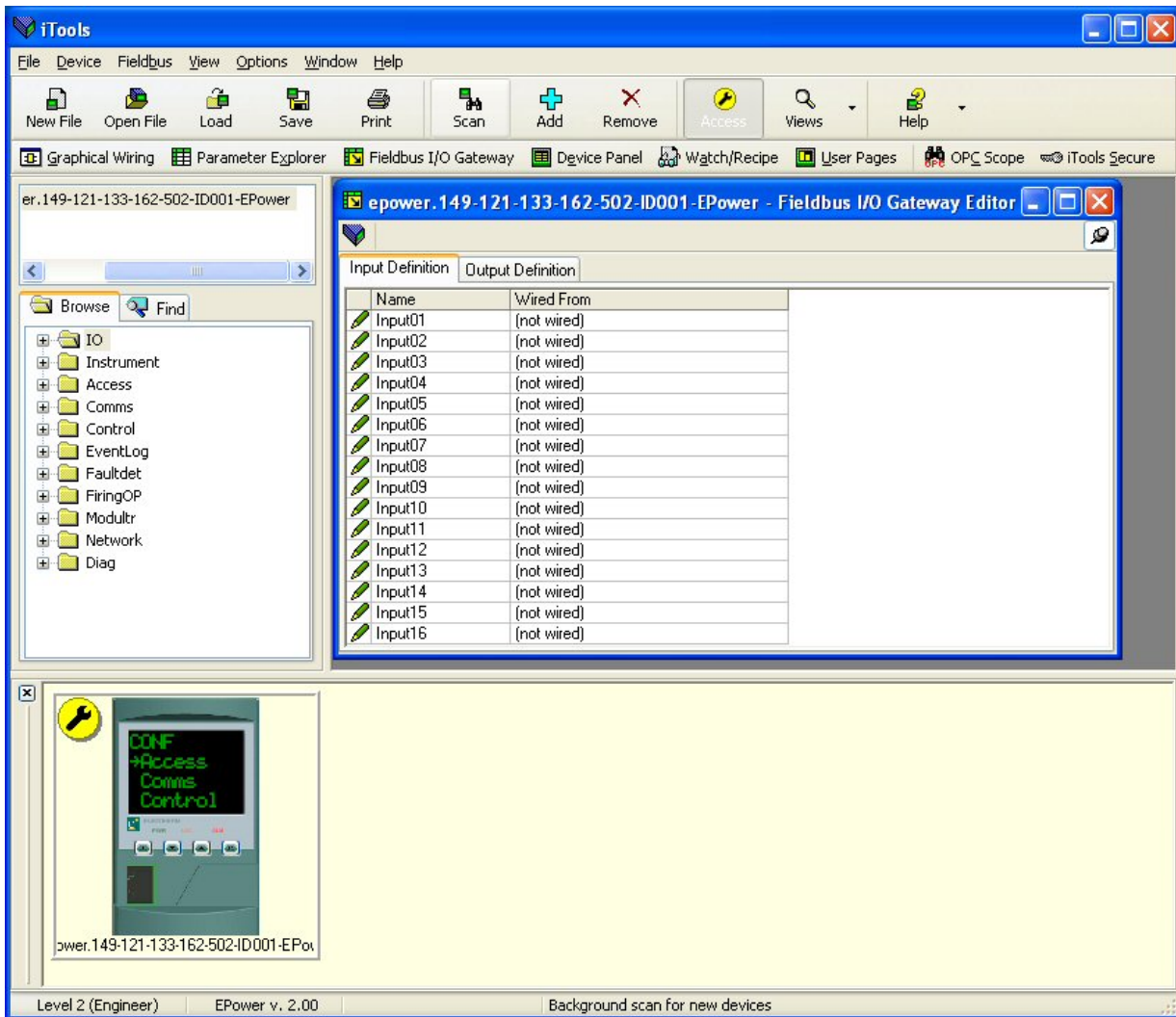


Figure 7-2: The I/O (Fieldbus I/O Gateway) Editor in iTools

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 7-3 shows an example of the pop-up and the input list produced.

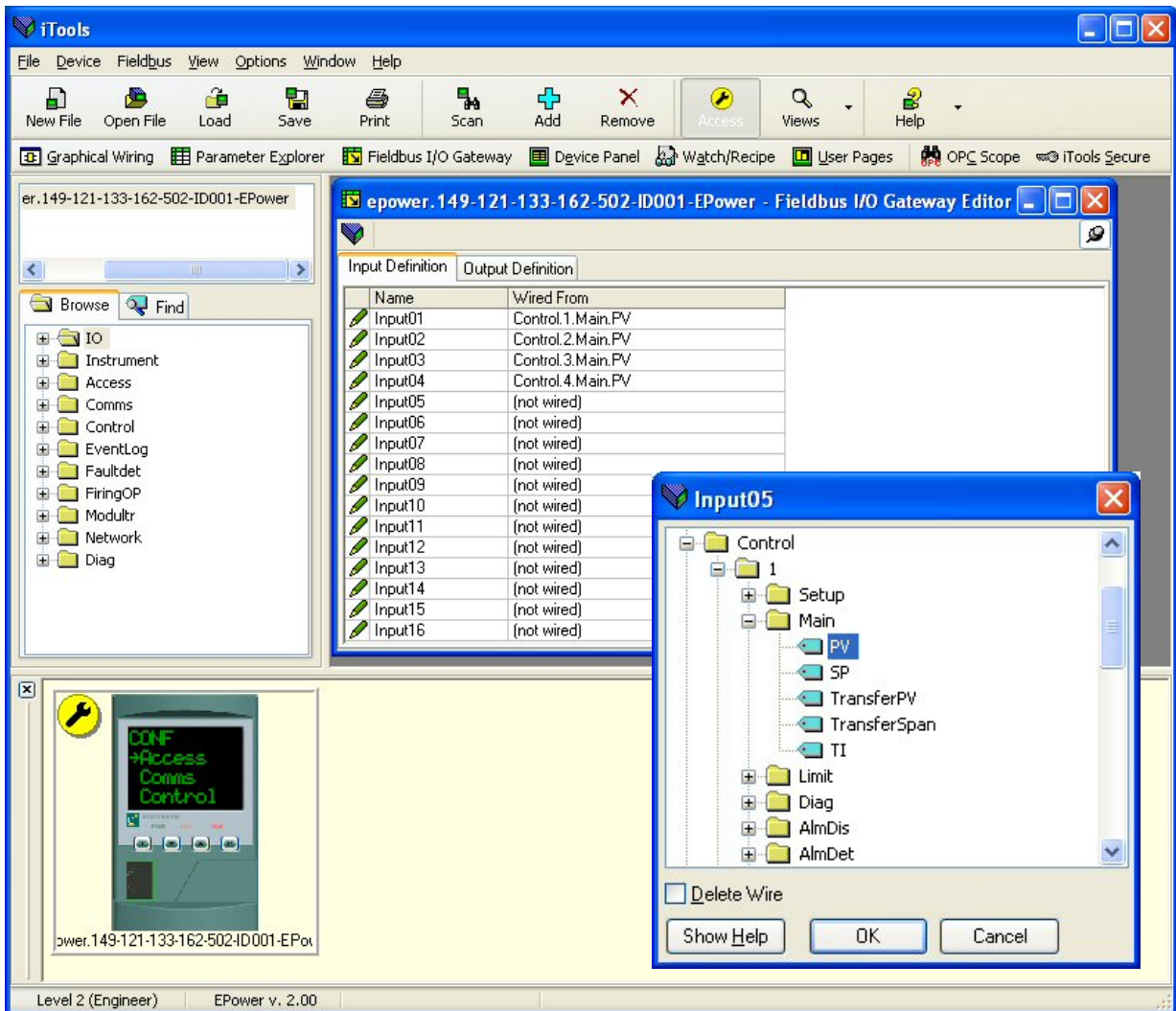



Figure 7-3: Selecting an Input Value and Example of an Input List

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 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 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.

7.6 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.co.uk or 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.

7.7 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.

7.8 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.

7.9 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. The 'Get Attribute Single' (0E_{hex}) and 'Set Attribute Single' (010_{hex}) services are used to retrieve and set values, in each case being applied to attribute 5 of the ADI object.

7.10 THE EDS FILE

The EDS (Electronic Data Sheet) file for EPower controller is named EPOWER.EDS and is available from your supplier, or electronically by going to Web site (www.eurotherm.com). 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.

7.11 TROUBLESHOOTING

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 section 1.1).
- 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.

8. APPENDIX A. 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

9. APPENDIX B. 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	i
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	4B	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	~
29)	54	T	7F	DEL
2A	*	55	U		

10. INDEX

- Access, 2, 29, 30, 34
- Acyclic, 2, 37, 42
- Address, 1, 2, 12, 13, 14, 15, 22, 23, 25, 32, 33, 35, 36, 39, 42, 45, 49
- Baud rate, 27, 37, 45, 48
- Block, 31
- Booleans, 31
- Cable, 1, 5, 9, 38
- Cables, 5
- Cat5, 34
- Command, 22, 23, 24, 25, 27
- Configuration, 2, 12, 30, 35, 45, 49
- Control signals, 6
- CRC, 1, 12, 15, 16, 17, 18, 19, 20, 22, 23, 24, 25, 26, 27, 50
- Cyclic, 1, 12, 15, 16, 37, 50
- Data exchange, 2, 40, 46
- Default gateway, 34
- Delay, 1, 12, 27
- DHCP, 2, 34, 35
- Diagnostic, 1, 8, 24
- Differential Mode, 4
- Drop lines, 44
- Duplex, 50
- Dynamic, 2, 35
- EDS, 2, 43, 48
- Enumerated parameters, 31
- EOT, 15, 27, 50, 51
- Error, 1, 13, 14, 23, 26, 28, 29
- Even, 12
- Fieldbus, 1, 28, 40, 41, 46
- Fixed, 35
- Floating point, 31
- Function code, 18, 21, 22, 23, 24, 25, 26
- Grounding, 1, 6
- GSD, 37, 42
- IEEE, 30, 31, 32, 33
- Impedance, 38
- Input, 40, 41, 42, 45, 46, 47, 48
- Integer, 2, 31
- Internet Site
 - UK, 11, 48
- IP address, 34, 35, 36
- iTools, 1, 2, 3, 11, 35, 36, 37, 39, 40, 45, 46
- KD485, 3, 4, 9, 10, 37
- Latency, 1, 12, 27
- Loopback, 1, 21, 24
- LSB, 17, 18, 22, 23, 24, 25, 26, 32, 33, 50
- MAC, 34, 35
- Master, 2, 35, 37, 41, 42, 47
- Message, 1, 15, 27, 50
- Mode, 1, 8, 15, 27, 29
- MSB, 17, 18, 22, 23, 24, 25, 26, 32, 33, 50
- Node, 44, 48, 49
- None, 14
- Odd, 12
- Output, 28, 40, 42, 46, 48
- Parity, 1, 12, 15, 50
- Patch cable, 10
- PLC, 37, 40, 46
- Protocol, 1, 2, 3, 12, 13, 14, 35
- Read, 1, 12, 21, 22, 31
- Receive, 9
- Register, 17, 18, 48
- Resolution, 1, 2, 14, 30
- RJ45, 6, 9, 10, 34
- Slave, 37, 42
- Start, 15, 50
- Status, 1, 2, 8, 28, 29, 31
- Stop, 15, 50
- Subnet mask, 34
- terminals, 9, 10, 42, 49
- Termination, 44
- Terminator, 10
- Time, 1, 2, 12, 27, 31, 33
- Transmit, 9
- Trunk, 44
- Twisted pair, 38
- Wait, 1, 14, 27
- Write, 1, 12, 21, 23, 25, 30, 31

Inter-Company sales and service locations

AUSTRALIA Sydney

Eurotherm Pty. Ltd.
Telephone (+61 2) 9838 0099
Fax (+61 2) 9838 9288
E-mail info.au@eurotherm.com

AUSTRIA Vienna

Eurotherm GmbH
Telephone (+43 1) 7987601
Fax (+43 1) 7987605
E-mail info.at@eurotherm.com

BELGIUM & LUXEMBOURG Moha

Eurotherm S.A./N.V.
Telephone (+32) 85 274080
Fax (+32) 85 274081
E-mail info.be@eurotherm.com

BRAZIL Campinas-SP

Eurotherm Ltda.
Telephone (+5519) 3707 5333
Fax (+5519) 3707 5345
E-mail info.br@eurotherm.com

DENMARK Copenhagen

Eurotherm Danmark AS
Telephone (+45 70) 234670
Fax (+45 70) 234660
E-mail info.dk@eurotherm.com

FINLAND Abo

Eurotherm Finland
Telephone (+358) 22506030
Fax (+358) 22503201
E-mail info.fi@eurotherm.com

FRANCE Lyon

Eurotherm Automation SA
Telephone (+33 478) 664500
Fax (+33 478) 352490
E-mail info.fr@eurotherm.com

GERMANY Limburg

Eurotherm Deutschland GmbH
Telephone (+49 6431) 2980
Fax (+49 6431) 298119
E-mail info.de@eurotherm.com

HONG KONG & CHINA

Eurotherm Limited North Point
Telephone (+85 2) 28733826
Fax (+85 2) 28700148
E-mail info.hk@eurotherm.com

Guangzhou Office

Telephone (+86 20) 8755 5099
Fax (+86 20) 8755 5831
E-mail info.cn@eurotherm.com

Beijing Office

Telephone (+86 10) 6567 8506
Fax (+86 10) 6567 8509
E-mail info.cn@eurotherm.com

Shanghai Office

Telephone (+86 21) 6145 1188
Fax (+86 21) 6145 1187
E-mail info.cn@eurotherm.com

INDIA Chennai

Eurotherm India Limited
Telephone (+9144) 2496 1129
Fax (+9144) 2496 1831
E-mail info.in@eurotherm.com

IRELAND Dublin

Eurotherm Ireland Limited
Telephone (+353 1) 4691800
Fax (+353 1) 4691300
E-mail info.ie@eurotherm.com

ITALY Como

Eurotherm S.r.l.
Telephone (+39 031) 975111
Fax (+39 031) 977512
E-mail info.it@eurotherm.com

KOREA Seoul

Eurotherm Korea Limited
Telephone (+82 31) 2738507
Fax (+82 31) 2738508
E-mail info.kr@eurotherm.com

NETHERLANDS Alphen a/d Rijn

Eurotherm B.V.
Telephone (+31 172) 411752
Fax (+31 172) 417260
E-mail info.nl@eurotherm.com

NORWAY Oslo

Eurotherm A/S
Telephone (+47 67) 592170
Fax (+47 67) 118301
E-mail info.no@eurotherm.com

POLAND Katowice

Invensys Eurotherm Sp z o.o
Telephone (+48 32) 218 5100
Fax (+48 32) 217 7171
E-mail info.pl@eurotherm.com

SPAIN Madrid

Eurotherm España SA
Telephone (+34 91) 661 6001
Fax (+34 91) 661 9093
E-mail info.es@eurotherm.com

SWEDEN Malmo

Eurotherm AB
Telephone (+46 40) 384500
Fax (+46 40) 384545
E-mail info.se@eurotherm.com

SWITZERLAND Wollerau

Eurotherm Produkte (Schweiz) AG
Telephone (+41 44) 787 1040
Fax (+41 44) 787 1044
E-mail info.ch@eurotherm.com

UNITED KINGDOM Worthing

Eurotherm Limited
Telephone (+44 1903) 268500
Fax (+44 1903) 265982
E-mail info.uk@eurotherm.com
Web www.eurotherm.co.uk

U.S.A Leesburg VA

Eurotherm Inc.
Telephone (+1 703) 443 0000
Fax (+1 703) 669 1300
E-mail info.us@eurotherm.com
Web www.eurotherm.com

ED54



invensys
EUROTHERM

EUROTHERM LIMITED

Faraday Close, Durrington, Worthing, West Sussex, BN13 3PL
Telephone: +44 (0)1903 268500 Facsimile: +44 (0)1903 265982
e-mail: info.uk@eurotherm.com
Website: <http://www.eurotherm.co.uk>

