SERIES 2000 COMMUNICATION HANDBOOK

Contents

Issue Status of this manual

Issue 3 adds the warning as follows:

⚠ **Warning. The 2200 and 3200 series makes use of EEPROM memory technology to retain variables and settings when the unit is powered down. This memory has a specified upper limit of, typically, 100,000 change cycles, and if this limit is exceeded, permanent damage will result to the memory and the instrument will display an error code and cease to operate.**

In the 2200 series retained parameters are written to EEPROM whenever they are changed using digital communications. It is important, therefore, that parameters which do not require updating on a regular basis (for example, setpoints, alarm trip levels, hysteresis, etc) are only written to when a change in the parameter value occurs.
It is also important when using digital communications or digital inputs to avoid writing constantly changing values to the instruments to ensure that fewer than 100,000 changes are not likely to any variable during the intended lifetime of the product.

In 3200 series parameters are only written to EEPROM when a change in its value occurs. However, it is still important to avoid writing constantly changing values.

If in doubt contact Eurotherm technical support for advice.

1. CHAPTER 1 INTRODUCTION

This chapter describes the scope of this handbook and how to use it.

1.1 OVERVIEW

This handbook is written for the people who need to use a digital communications link and MODBUS, JBUS, or EI-BISYNCH communication protocols to supervise Eurotherm Controls Series 2000 instruments, including the 2200 and 2400 instrument models.

It has been assumed that the reader has some experience of communication protocols and is familiar with Series 2000 instruments. The relevant instrument handbook gives a full description of how to use the instruments, configuration options and definition of parameters.

Chapter 2 of this document is a guide to cabling and the basic physical environment of digital communications.

Chapter 3 is a general description of the MODBUS and JBUS protocols.

Chapter 4 is a general description of the EI-BISYNCH protocol.

Chapter 5 lists Series 2000 parameter addresses and mnemonics.

Chapter 6 covers advanced topics such as access to full resolution floating point data and user interface permissions.

Appendix A is a Glossary of Terms.

Appendix B lists ASCII codes.

Eurotherm Controls accepts no responsibility for any loss or damage caused by application of the information contained in this document.

JBUS ® is a registered trademark of APRIL.
MODBUS ® is a registered trademark of Gould Inc.

1.2 JBUS V MODBUS

- MODBUS is a serial communications protocol defined by Gould Inc.
  April developed JBUS as a special case of MODBUS.
- The two protocols use the same message frame format.
- The function codes used by Series 2000 instruments are a subset of JBUS and MODBUS function codes.
- NB. Series 2000 JBUS addresses are exactly the same as MODBUS addresses.
  This differs from previous implementations of the protocol in Eurotherm equipment.
- In this document reference will be made to MODBUS, however all information applies equally to JBUS.

1.3 REFERENCES

Refer to the documents below for further information;

Gould	MODBUS Protocol Reference Guide, PI-MBUS-300
April	JBUS Specification
EIA Standard RS-232-C	Interface Between Terminal Equipment and Data Communication Equipment Employing Serial Binary Interchange
EIA Standard RS-422	Electrical Characteristics of Balanced Voltage Digital Interface Circuits
EIA Standard RS-485	Electrical Characteristics of Generators and Receivers for use in Balanced Digital Multipoint Systems

2. CHAPTER 2 DIGITAL COMMUNICATIONS HARDWARE

This chapter defines the differences between the EIA232, EIA422 and EIA485 digital communications standards. EIAxxx is also known as RSxxx.

Details of configuration, cabling and termination will help to establish basic communications.

2.1 EIA232, EIA422 AND EIA485 TRANSMISSION STANDARDS

The Electrical Industries Association, (EIA) introduced the Recommended Standards, EIA232, EIA422 and EIA485. 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	EIA422	EIA485
Transmission mode	Single ended	Differential	Differential
Electrical connections	3 wire	5 wire	3 wire
No. of drivers and receivers per line	1 driver, 1 receiver	1 driver, 10 receivers	32 drivers, 32 receivers
Maximum data rate	20k bits/s	10M bits/s	10M bits/s
Maximum cable length	50ft, (15M)	4000ft, (1200M)	4000ft, (1200M)

Note: EIA232C has been abbreviated to EIA232. 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 two wire connection, with cable length of less than **1200M**. 31 Instruments and one 'master' may be connected in this way. The balanced differential signal transmission is less prone to interference and should be used in preference to EIA232 in noisy environments. EIA422/ EIA485 is recommended for plant installation. Although EIA485 is commonly referred to as a 'two wire' connection, a ground return/shield connection is provided as a 'common' connection for Series 2000 Instruments, and in general this should be used in installations to provide additional protection against noise.

Strictly speaking, EIA422 is a standard permitting 'point to point' connection of two pieces of equipment using a full duplex, differential signal on two pairs of wires. In principle, therefore, an EIA422 link only allows a single instrument to be connected to a PC. However, Series 2000 instruments provide an enhanced version of EIA422 that also meets the full electrical requirements of EIA485 described above. This allows up to 31 instruments to be connected on the same network, but only with a **5 wire** electrical connection. The transmission and reception of data use two pairs of twisted cable, with a separate cable provided for common. The optional screen will provide additional noise immunity.

The 2 wire EIA485 should be used where possible for new installations where multi-drop capability is required. EIA422 is provided for compatibility with existing Eurotherm instruments.

Some earlier Eurotherm Controls instruments use a terminal marking that is different from that used in the EIA422/ EIA485 standards. The table below compares this marking.

EIA422/ EIA485	Eurotherm
A'	RX+
B'	RX-
A	TX+
B	TX-
C & C'	Common

Using EIA232 or EIA422/485, the Series 2000 instruments 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 Eurotherm Controls KD485 Comms Interface unit is recommended for interfacing to EIA422/485. This unit is also used to buffer an EIA422/485 network when it is required to communicate with more than 32 instruments on the same bus, and may also be used to bridge 2 wire EIA485 to 4 wire EIA422 network segments. Wiring information for this unit is given at the end of this chapter, or refer to KD485 Installation and Operation Manual, available on request from Eurotherm, for a full description.

2.2 SELECTING EIA232 OR EIA422/485

Changing between EIA232, EIA422, and EIA485 is possible for 2400 Series instruments by replacing the plug-in 'H' Module with a communications module of the required type.

2200 Series communications hardware is a fixed build and must be specified when the instrument is being ordered.

2.3 CABLE SELECTION

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 earth).
- For EIA422/485 applications, use twisted pair cables.

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 earth). For applications communicating over longer distances, choose a cable that also has low attenuation characteristics.

In low noise applications and over short distances it may be possible to use the earthed screen as the common connection. Connect the common to the earthed screen via a 100 ohm, 1/4W carbon composition resistor at the PC and all instruments.

For EIA422/485 it is possible to operate the system with unscreened twisted data pairs, earth is used as the common connection. Connect the common to earth via a 100 ohm, 1/4W carbon composition resistor at the PC and all instruments. This system is not recommended.

The following list is a selection of cables suitable for EIA422/485 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.

Part number	
Belden	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 *

The following are a selection of cables suitable for EIA232 communication systems listed in order of decreasing quality;

Part number		
Alpha	Belden	Description
	8102	2 twisted pairs with aluminium foil screen plus a 65% coverage copper screen **
5472	9502	2 twisted pairs with aluminium foil screen *
2403	8771	3 separate wires with aluminium foil screen **

2.4 EARTHING (GROUNDING)

Ensure all earth (ground) points are noise free.

To reduce interference from external electrical signals, earth the cable screen at a single earth point. There must not be multiple ground paths in a single cable run. When using a Eurotherm Controls KD485 Communications Adapter unit, do not connect the screen from one side of the interface to the other. Rather, ground each of the cables separately at a local ground point.

The digital communication outputs of all Series 2000 instruments are isolated. To avoid common mode noise problems, connect the common line to ground at one point through a 100 ohm, 1/4W, carbon composition resistor. The resistor will limit the ground current.

2.5 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.

2.6 PRECAUTIONS

2000 series discrete instruments 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.

2.7 WIRING EIA232

To use EIA232 the PC will be equipped with an EIA232 port, usually referred to as COM 1.

To construct a cable for EIA232 operation use a three core screened cable.

The terminals used for EIA232 digital communications are listed in the table below. Some PC's use a 25 way connector although the 9 way is more common.

Standard Cable	PC socket pin no.		PC Function *	Instrument Terminal	Instrument
Colour	9 way	25 way			Function
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		

- These are the functions normally assigned to socket pins. Please check your PC manual to confirm.

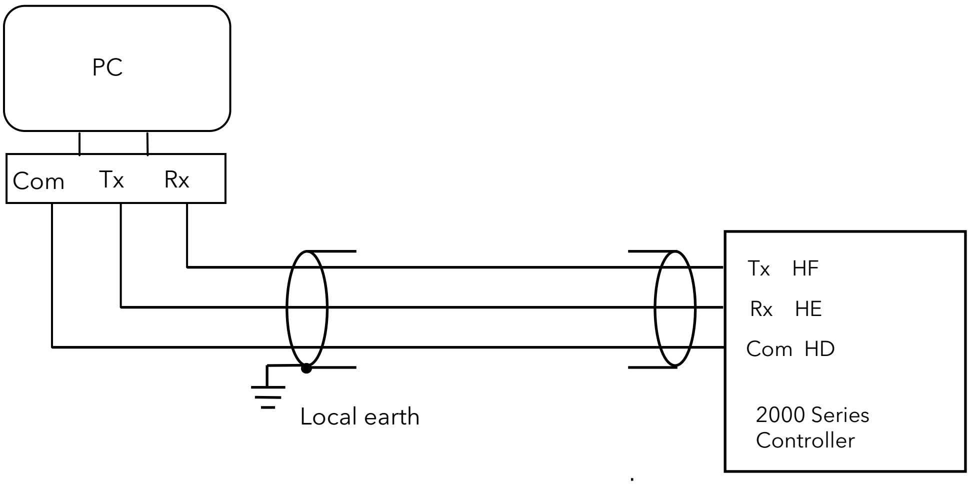


Figure 2-1: EIA232 Connections

2.8 WIRING EIA422 OR 4-WIRE EIA485

To use EIA422, buffer the EIA232 port of the PC with a suitable EIA232/ EIA422 converter. The Eurotherm Controls KD485 or 261 Communications Converter unit is recommended for this purpose. Figure 2-5 shows connections for this unit. Instruments on a EIA422 communication network should be chain connected and not star connected.

To construct a cable for EIA422 operation use a screened cable with two twisted pairs plus a separate core for common. Although common or screen connections are not necessary, their use will significantly improve noise immunity.

The terminals used for EIA422 digital communications are listed in the table below.

Standard Cable	PC socket pin no.	PC Function *	Instrument Terminal		Instrument
Colour	25 way		902-4	2400	Function
White	3	Receive (RX+)	F1	HE	Transmit (TX+)
Black	16	Receive (RX-)	F2	HF	Transmit (TX-)
Red	12	Transmit (TX+)	F3	HB	Receive (RX+)
Black	13	Transmit (TX-)	F4	HC	Receive (RX-)
Green	7	Common	F5	HD	Common
Screen	1	Ground			

- These are the functions normally assigned to socket pins. Please check your PC manual to confirm.

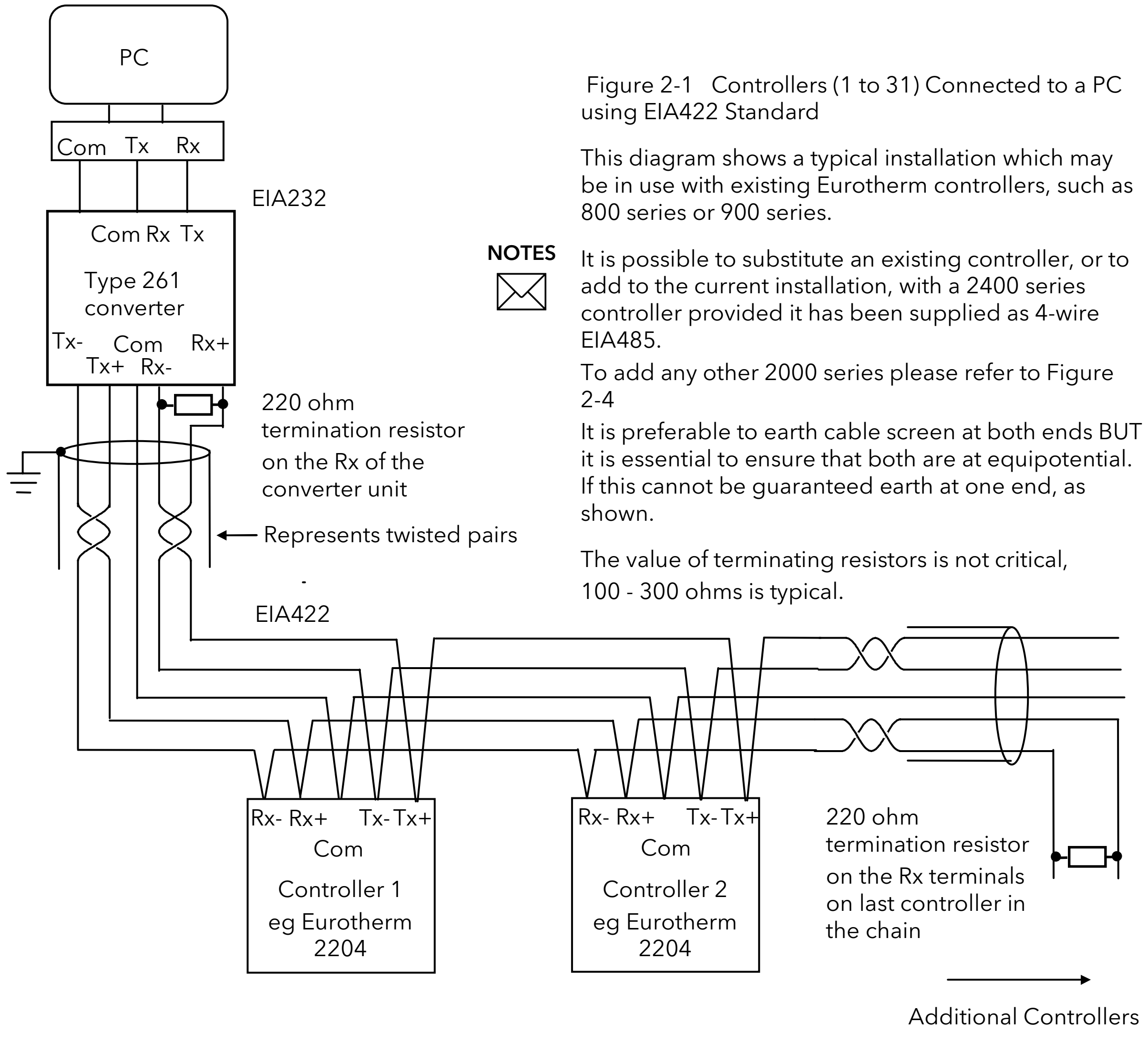


Figure 2-1 Controllers (1 to 31) Connected to a PC using EIA422 Standard

This diagram shows a typical installation which may be in use with existing Eurotherm controllers, such as 800 series or 900 series.

NOTES

It is possible to substitute an existing controller, or to add to the current installation, with a 2400 series controller provided it has been supplied as 4-wire EIA485.

To add any other 2000 series please refer to Figure 2-4

It is preferable to earth cable screen at both ends BUT it is essential to ensure that both are at equipotential. If this cannot be guaranteed earth at one end, as shown.

The value of terminating resistors is not critical, 100 - 300 ohms is typical.

Figure 2-2: EIA485 Connections

2.9 WIRING 2-WIRE EIA485

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. Eurotherm does not recommend the use of an EIA485 board built into the computer since this board is unlikely to be isolated, which may cause noise problems, and the Rx terminals are unlikely to be biased correctly for this application.

To construct a cable for EIA485 operation use a screened cable with one (EIA485) twisted pair plus a separate core for common. Although common or screen connections are not necessary, their use will significantly improve noise immunity.

The terminals used for EIA485 digital communications are listed in the table below.

Standard Cable Colour	PC socket pin no. 25 way	PC Function *	Instrument Terminal	Instrument Function
White	3	Receive (RX+)	HF (b) or (B+)	Transmit (TX)
Black	16	Receive (RX-)		
Red	12	Transmit (TX+)	HE (A) or (A+)	Receive (RX)
Black	13	Transmit (TX-)		
Green	7	Common	HD	Common
Screen	1	Earth		

* These are the functions normally assigned to socket pins. Please check your PC manual to confirm .

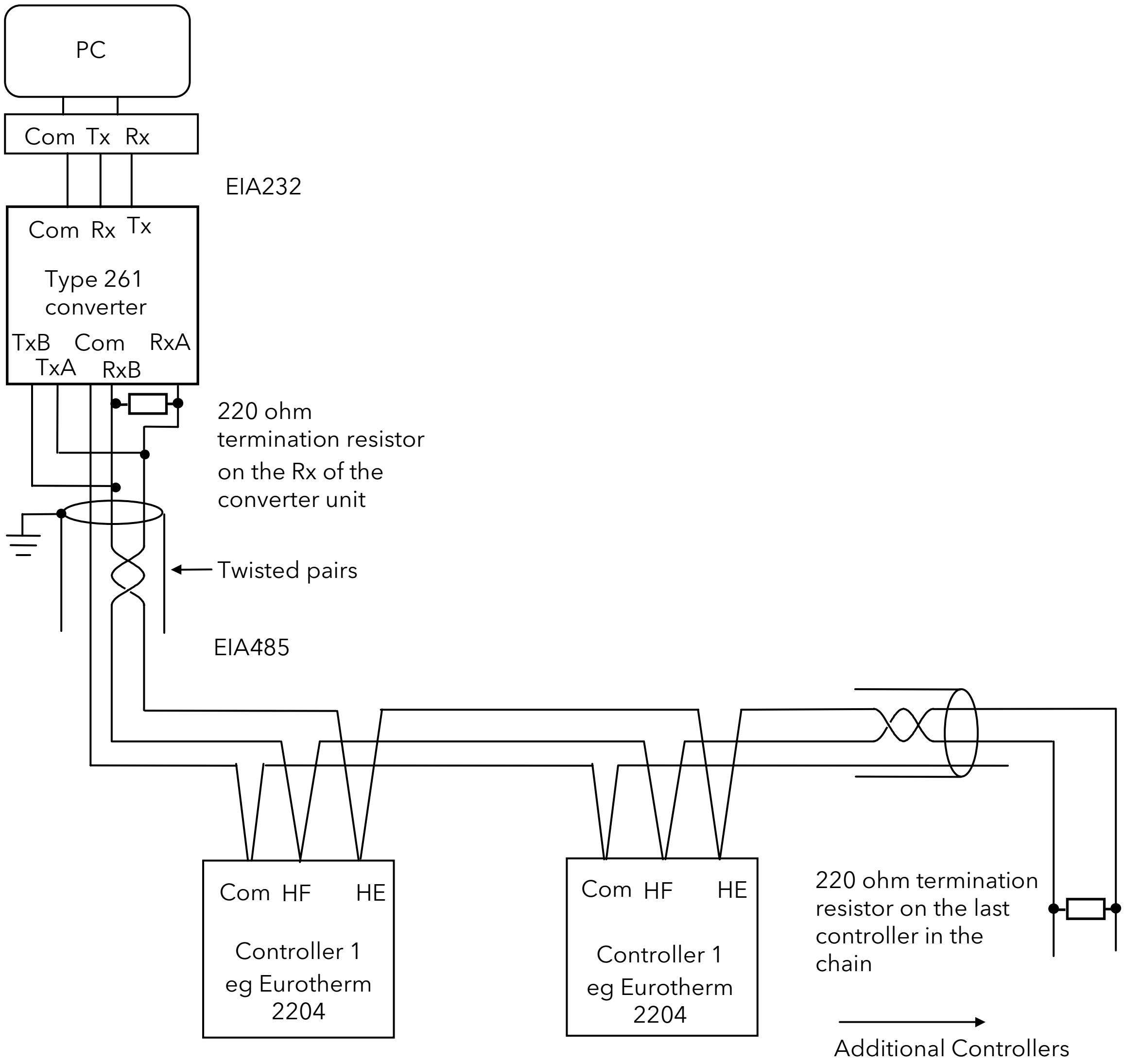

PC
Com Tx Rx
EIA232
Com Rx Tx
Type 261 converter
TxB Com RxA
TxA RxB
220 ohm termination resistor on the Rx of the converter unit
Twisted pairs
EIA485
Com HF HE
Controller 1 eg Eurotherm 2204
Com HF HE
Controller 1 eg Eurotherm 2204
220 ohm termination resistor on the last controller in the chain
Additional Controllers

Figure 2-3: 2000 series controllers (1 to 31) connected to a PC using 2-wire EIA485 standard

2.10 WIRING EIA422 AND EIA485 CONTROLLERS

It is generally not possible to mix controllers using a 2-wire and 4-wire standards. However, in some cases it may be required to add new controllers (such as 2000 series on 2-wire comms) to an existing installation with controllers on 4-wire comms.

In these cases the existing communications link may be modified by adding a special version of the KD485 converter unit, supplied as KD485-ADE 422-422. Figure 2-4 shows a large system of controllers using a mixture of both standards.

The standard KD485 unit converts from 232 to 4-wire 485 and this link is used to communicate to the first 32 existing controllers. The second KD485 is the special version which converts from 4-wire to 4-wire communications to communicate with a further bank of 32 controllers on 4-wire communications. The third KD485 is the same special version which is connected to convert from 4-wire to 2-wire communications. It's input side behaves to the 4-wire link as another controller would on an existing system, whilst at the same time the communications messages from the computer are passed onto the output side of this unit.

This is connected to the 2-wire communications link, that will contain the series 2000 controllers. Any responses from controllers on this link will cause data to be placed on to the 4-wire link and thence will be passed back to the computer.

The comms connections required for large systems are a mixture of 2-wire and 4-wire comms. Controllers 1 to 31 are 4-wire, Controllers 32 to 64 are 4-wire, controllers 65 to 97 are 2-wire.

Notes:

It is possible to connect 4-wire to 2-wire. It is not possible to connect 2-wire to 4-wire.

If it is necessary to configure the KD485 422-422, please note that an additional 232 to 422 converter is necessary for this.

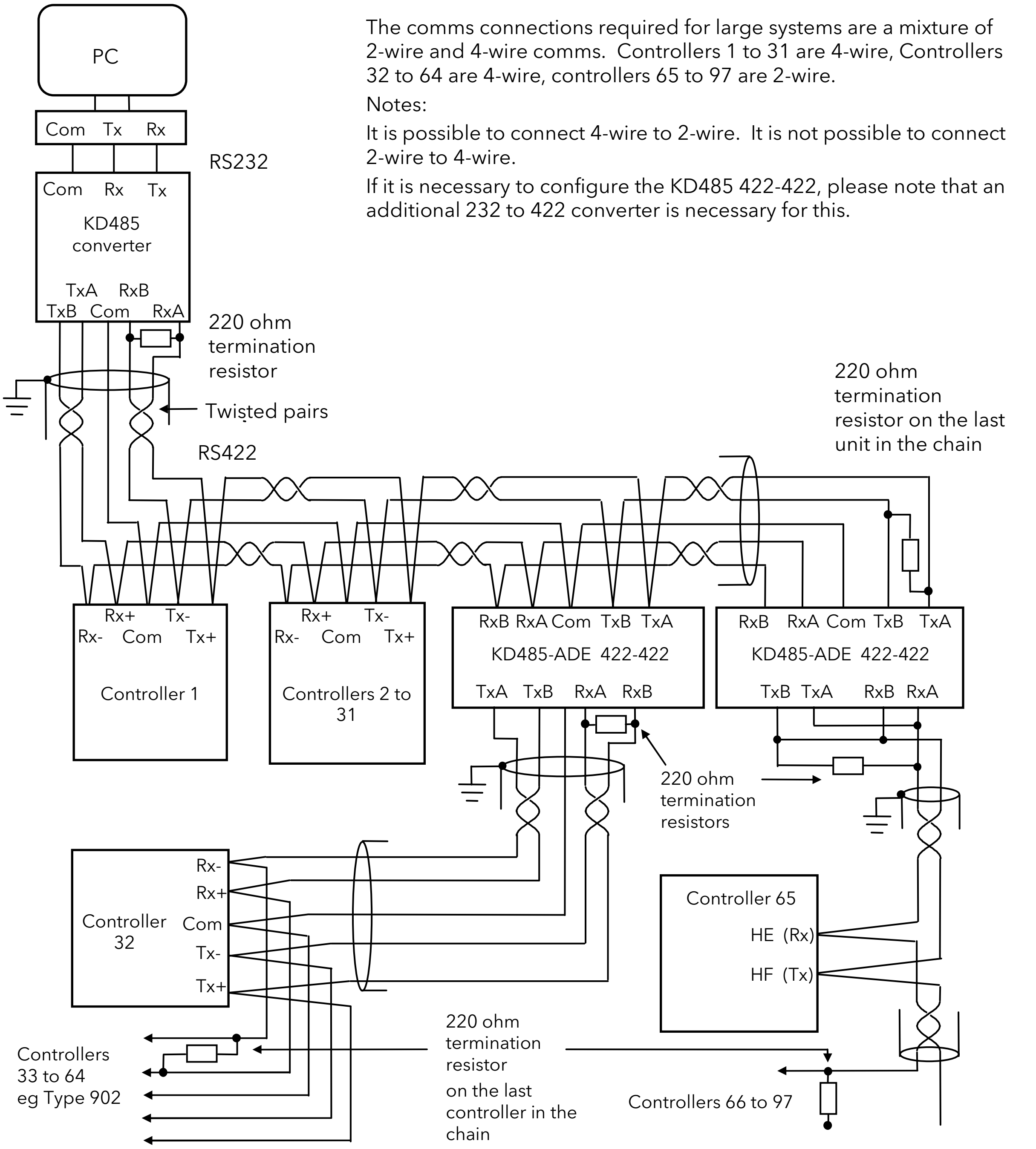


Figure 2-4: Controllers (1 to 31) connected to a PC using a mixed standard of EIA422 (or EIA485 4-wire) and EIA485 2-wire

2.11 LARGE EIA422/485 NETWORKS

Networks with more than 32 instruments will require buffering of the communication lines. The Eurotherm KD485-ADE 422-422, Universal Serial Interface unit is recommended for this purpose.

The KD485 in this format sets the Transmit line's to non tristate.

NOTE Large networks using EIA422 4-wire controllers could use the Eurotherm 261 Universal Serial Interface Unit. To set the transmit lines to non tristate in 261, change links 4 and 5 from position B to A.

Contact Eurotherm for further information when specifying large networks.

Instruments on a EIA422/485 communication network should be chain connected and not star connected.

The diagram below illustrates the wiring of a network for communicating with a large number of 2000 Series controllers.

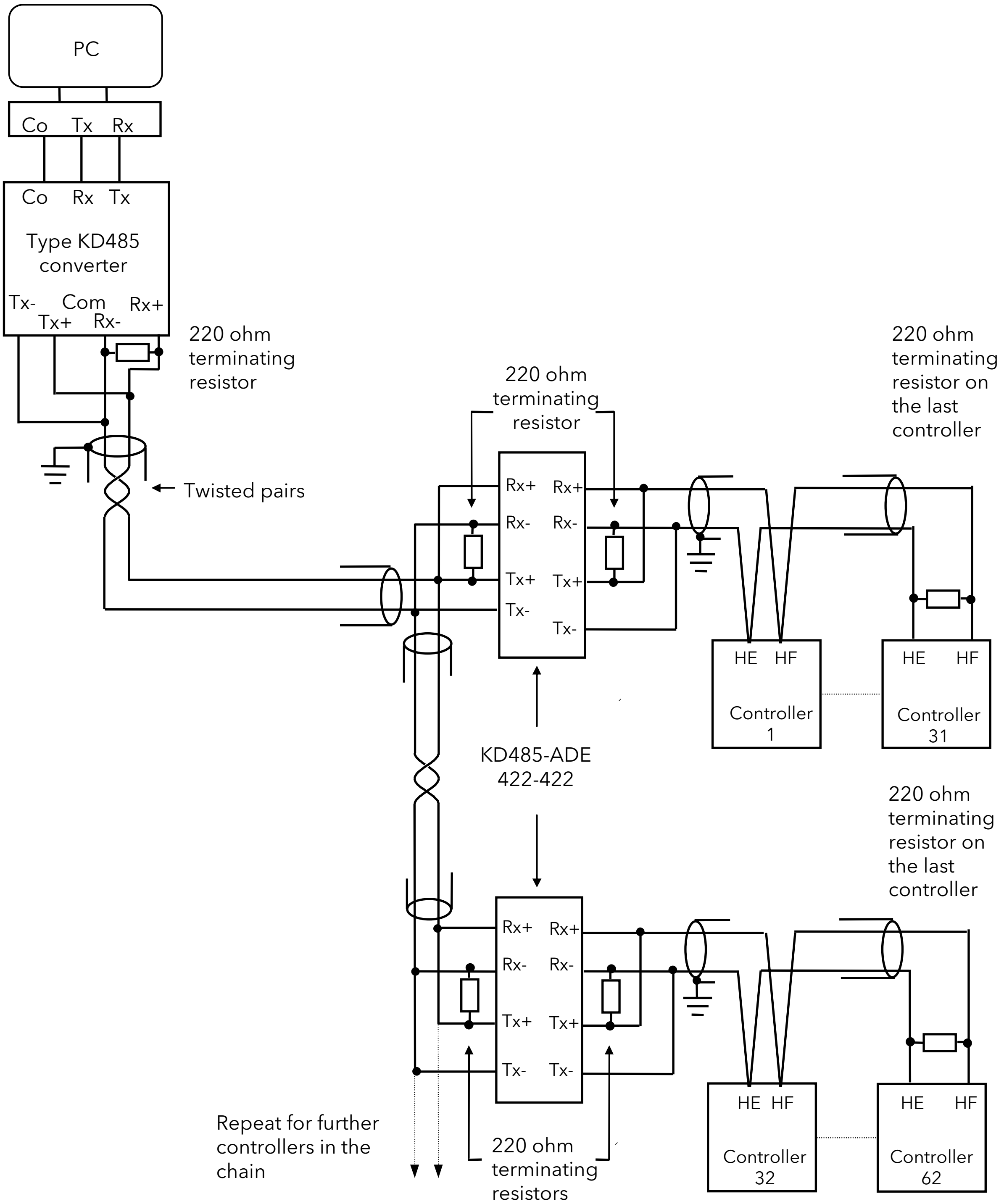

PC
Co Tx Rx
Co Rx Tx
Type KD485 converter
Tx- Com Rx+
Tx+ Rx-
220 ohm terminating resistor
Twisted pairs
220 ohm terminating resistor
220 ohm terminating resistor on the last controller
Rx+ Rx+
Rx- Rx-
Tx+ Tx+
Tx- Tx-
HE HF
Controller 1
HE HF
Controller 31
KD485-ADE 422-422
220 ohm terminating resistor on the last controller
Rx+ Rx+
Rx- Rx-
Tx+ Tx+
Tx- Tx-
HE HF
Controller 32
HE HF
Controller 62
Repeat for further controllers in the chain
220 ohm terminating resistors

Figure 2-5: Connections for large EIA422/485 Networks

2.12 EIA232 CONNECTIONS OF THE 261

Standard Cable	PC socket pin no.		PC Function *	261	
Colour	9 way	25 way		Terminal	Function
White	2	3	Receive (RX)	2	Transmit (TX)
Black	3	2	Transmit (TX)	3	Receive (RX)
Red	5	7	Common	7	Common
Screen		1	Earth		

* These are the functions normally assigned to socket pins. Please check your PC manual to confirm.

2.13 EIA422/485 CONNECTIONS OF THE 261

Standard Cable	PC socket pin no.	PC Function *	261	
Colour	25 way		Terminal	Function
White	3	Receive (RX+)	12	Transmit (TX+)
Black	16	Receive (RX-)	13	Transmit (TX-)
Red	12	Transmit (TX+)	3	Receive (RX+)
Black	13	Transmit (TX-)	16	Receive (RX-)
Green	7	Common	7	Common
Screen	1	Earth		

* These are the functions normally assigned to socket pins. Please check your PC manual to confirm.

2.14 CONNECTIONS FOR THE KD485-ADE

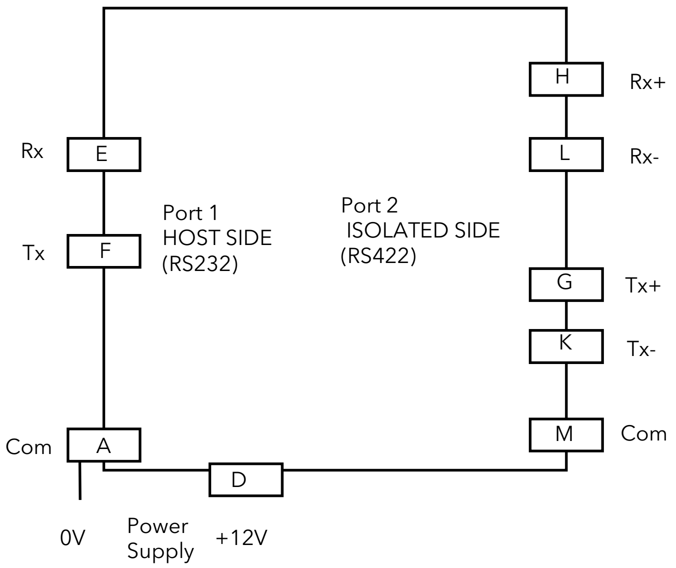


Figure 2-6: KD485 Connections

Further details are available in KD485 Installation and Operation Handbook.

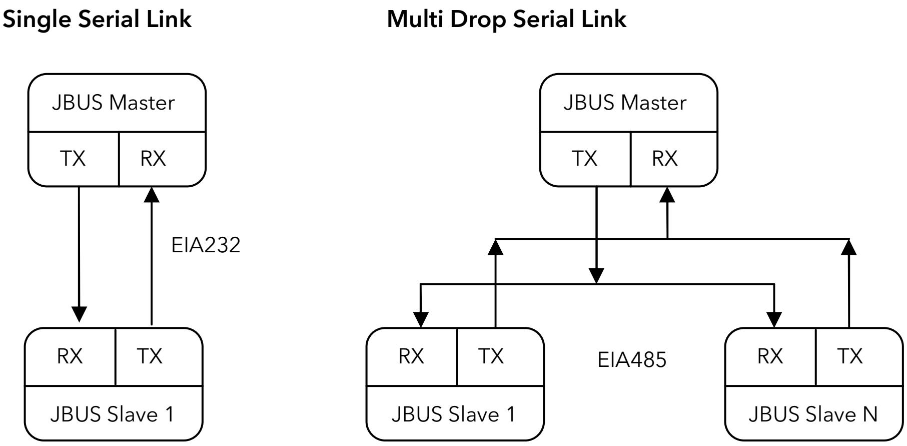
3. CHAPTER 3 MODBUS AND JBUS PROTOCOL

This chapter introduces the principles of the MODBUS and JBUS communication protocols. Note that in the Series 2000, these two protocols are identical, and 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;



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.

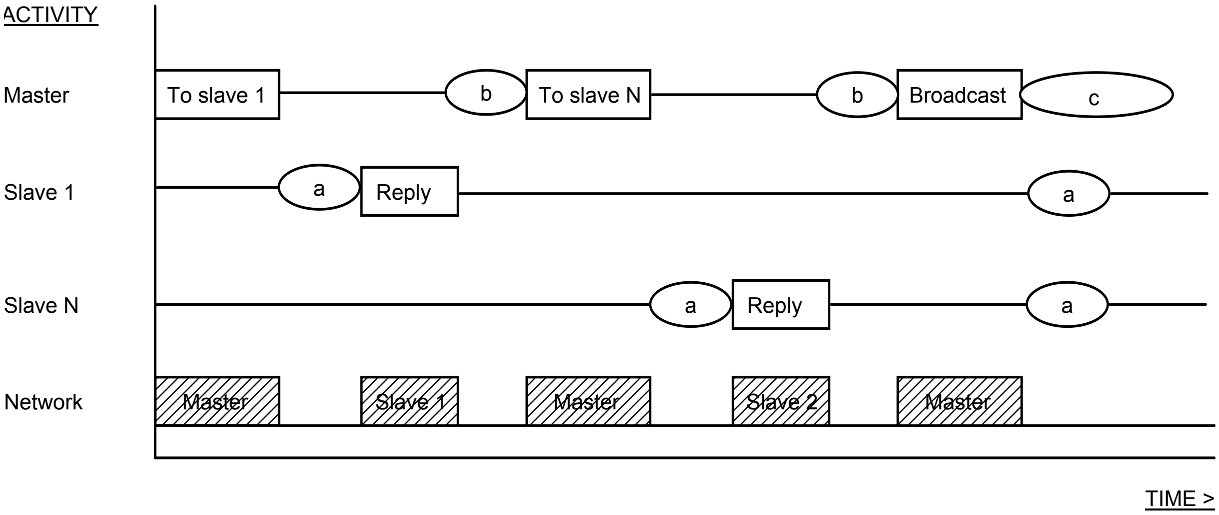
The message in either direction will consist of the following information;

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.
- Series 2000 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.

For a definition of the time periods required by the network, refer to 'Wait Period' in section 3.20 'Error Response'.

3.3 DEVICE ADDRESS

Each slave has a unique 8 bit device address. The Gould MODBUS Protocol defines the address range limits as 1 to 247. Series 2000 instruments will support an address range of 1 to 254. The device address used by the instrument is set using the Addr parameter in the CmS LiSt, which is available in operator mode. Note that this list may only be accessible when using the FuLL user interface: refer to the manual supplied with the instrument for more details on how to set this parameter.

Device address 0 is a special case that will broadcast a message to all slave devices simultaneously.

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 definitions for Series 2000 instruments are in Chapter 5.

3.5 PARAMETER RESOLUTION

JBUS and MODBUS protocol limit data to 16 bits per parameter. This reduces the active range of parameters to 65536 counts. In Series 2000 instruments this is implemented as -32767 (8001h) to +32767 (7FFFh).

The protocol is also limited to integer communication only. Series 2000 instruments allow the user to configure either integer or full resolution. In integer mode all parameters will be rounded to the nearest integer value, whereas 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.

The Series 2000 instruments provide a special sub-protocol for accessing full resolution floating point data. This is described in Chapter 6 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 JBUS and MODBUS Protocols define a mode of transmission for both ASCII and RTU modes of transmission. Series 2000 instruments **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 Series 2000 instruments 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:

Start	d7	d6	d5	d4	d3	d2	d1	d0	Parity	Stop

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 JBUS and MODBUS:

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. Series 2000 instruments 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 (⊕) 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.

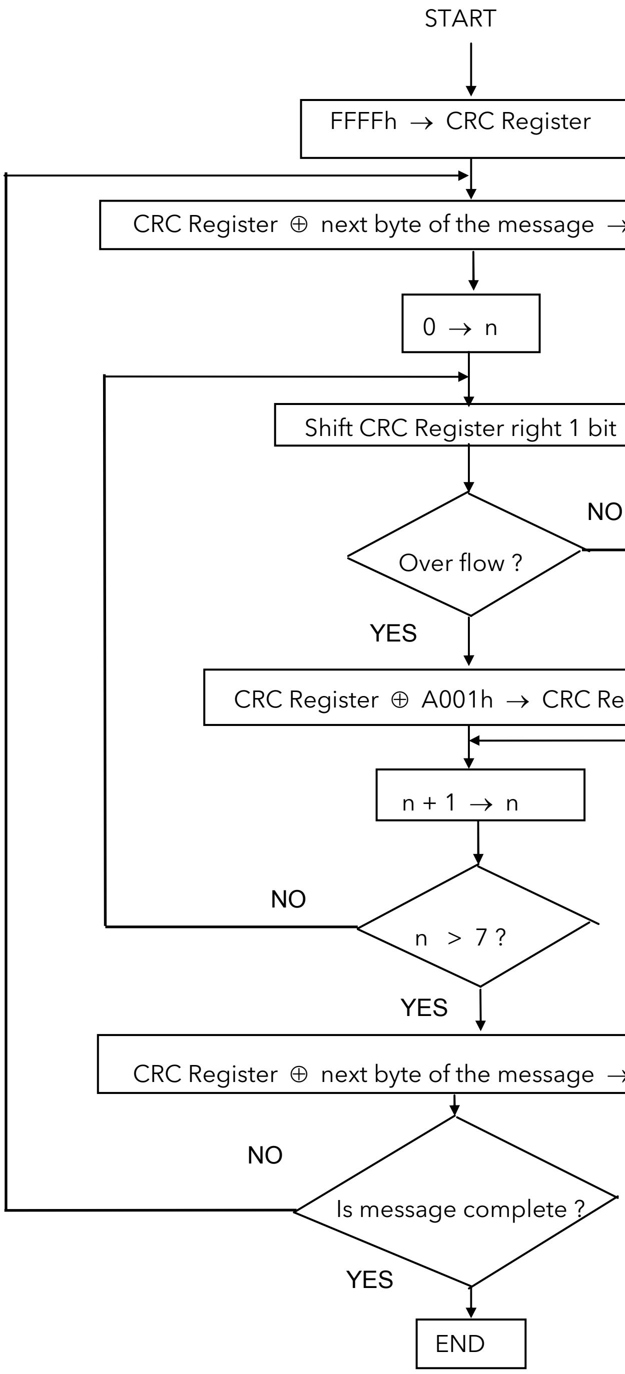
6 Exclusive OR the next 8 bit byte of the message with the high order byte of the CRC register.

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 '⊕' symbol indicates an 'exclusive OR' operation. 'n' is the number of data bits.


START
FFFFh → CRC Register
CRC Register ⊕ next byte of the message → CRC Register
0 → n
Shift CRC Register right 1 bit
Over flow ?
NO
YES
CRC Register ⊕ A001h → CRC Register
n + 1 → n
n > 7 ?
NO
YES
CRC Register ⊕ next byte of the message → CRC Register
Is message complete ?
NO
YES
END

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 communication functions are supported by Series 2000 instruments:

Function code	Function
01 or 02	Read n bits
03 or 04	Read n words
05	Write a bit
06	Write a word
07	Fast Read of Status
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.

Only the write function codes 05, 06 and 16 will work with a 'broadcast mode' address. ***2000 Series instruments will not reply if they receive a request including a unsupported function code.***

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

Parameter definitions for the Series 2000 instruments are provided later in this document.

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

3.13 READ N BITS

Function code: 01 or 02, (01h or 02h)

Command:

Device address	Function code 01 or 02	Address of first bit		Number of bits to read		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Reply:

Device address	Function code 01 or 02	Number of bytes read	First byte of data	....	Last byte of data	CRC	
1 byte	1 byte	1 byte	1 byte	....	1 byte	MSB	LSB

The first data byte contains the status of the first 8 bits, with the least significant bit being the first bit. The second data byte contains the status of the next 8 bits, etc. Unused bits are set to zero.

Example: From the instrument at device address 19, read 14 bits, beginning at parameter address 2.

Command:

Device address	Function code	Address of first bit		Number of bits to read		CRC	
13	01	00	02	00	0E	1F	7C

Reply:

Device address	Function code	Number of bytes read	First byte of data	Second byte of data	CRC	
13	01	02	01	01	C1	AF

An expansion of the data bytes illustrates the relationship between data and the parameter addresses. The reply indicates that the instrument is in sensor break and manual mode.

Data byte	1st byte (40h)								2nd byte (02h)							
Param. address	9	8	7	6	5	4	3	2	17	16	15	14	13	12	11	10
Bit values	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Parameter addresses 16 and 17 are set to zero.

3.14 READ N WORDS

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 for 2400 Series instruments and 32 for the 2200

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 2000 Series slave at device address 2, read 2 words from parameter address 1 (Process Variable and Target Setpoint).

Command:

Device address	Function code	Address of first word		Number of words to read		CRC	
02	03	00	01	00	02	95	F8

Reply: (If the instrument is configured with **integer** 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	12	00	16	E8	F8

Reply: (If the instrument is 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	B2	00	D8	69	4E

As the decimal point is not transmitted, the master must scale the response; 183=5.0, 216=10.0.

3.15 WRITE A BIT

Function code: 05, (05h)

Command:

Device address	Function code 05	Address of bit		Value of bit		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The LSB of 'Value of bit' is always set to 00. The MSB is used to write the value of the addressed bit.
To set a bit value of 1, either transmit 01h or FFh. To set a bit value of 0 transmit 00h.

A device address 00 will broadcast the data to all devices on the network.

Reply: (There will be no reply to a command broadcast to the device address 00.)

Device address	Function code 05	Address of bit		Value of bit		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

The reply to function 05 is the same as the command. See the section on 'Error Response' below for details of the reply if the operation fails.

Example: Write to the Series 2000 instrument at device address 2 and set the instrument to manual. (The bit at parameter address 2 is set).

Command:

Device address	Function code	Address of bit		Value of bit		CRC	
02	05	00	02	01	00	6D	A9

Reply:

Device address	Function code	Address of bit		Value of bit		CRC	
02	05	00	02	01	00	6D	A9

3.16 WRITE A WORD

Function code: 06, (06h)

Command:

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

A device address 00 will broadcast the data to all devices on the network.

Reply: (There will be no reply to a command broadcast to the device address 00.)

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

The reply to function 06 is the same as the command. See the section on ‘Error Response’ below for details of the reply if the operation fails.

Example: Write to the Series 2000 slave at device address 2 and change the setpoint to 25.0°C (address 2). 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	00	02	00	FA	A8	7A

Reply:

Device address	Function code	Address of word		Value of word		CRC	
02	06	00	02	00	FA	A8	7A

3.17 FAST READ OF STATUS

Function code: 07, (07h)

The fast read of status command is short to allow a rapid transaction to obtain one byte of frequently needed status information.

Command

Device address	Function code 07	CRC	
1 byte	1 byte	MSB	LSB

Reply:

Device address	Function code 07	Fast read status byte	CRC	
1 byte	1 byte	1 byte	MSB	LSB

The table below defines the status byte information used by Series 2000 instruments.

Parameter	**Modbus 2400**	**Bisynch 2400**	**Modbus 2200**	**Bisynch 2200**	**Display**
Summary Output Status Word	75	SO	75	SO	-

BIT	**DESCRIPTION**	
0	Alarm 1 State (0 = Safe, 1 = Alarm)	Alarm 1 State (0 = Safe, 1 = Alarm)
1	Alarm 2 State (0 = Safe, 1 = Alarm)	Alarm 2 State (0 = Safe, 1 = Alarm)
2	Alarm 3 State (0 = Safe, 1 = Alarm)	Alarm 3 State (0 = Safe, 1 = Alarm)
3	Alarm 4 State (0 = Safe, 1 = Alarm)	Alarm 4 State (0 = Safe, 1 = Alarm)
4	Manual Mode (0 = Auto, 1 = Manual)	Manual Mode (0 = Auto, 1 = Manual)
5	Sensor Break (0 = Good PV, 1 = Sensor Broken)	Sensor Break (0 = Good PV, 1 = Sensor Broken)
6	Loop Break (0 = Good closed loop, 1 = Open Loop)	Loop Break (0 = Good Closed Loop, 1 = Open Loop)
7	Heater Fail (0 = No Fault, 1 = Load fault detected)	Heater Fail (0 = No Fault, 1 = Load Fault Detected)
8	Tune Active (0 = Auto Tune disabled, 1 = Auto Tune active)	Load Fail (0 = No Fault, 1 = Load Fault Detected)
9	Ramp/Program Complete (0 = Running/Reset, 1 = Complete)	Ramp/Program Complete (0 = Running/Reset, 1 = Complete)
10	PV out of range (0 = PV within table range, 1 = PV out of table range)	PV out of range (0 = PV within table range, 1 = PV out of table range)
11	DC control module fault (0= Good,. 1= BAD)	SSR Fail (0 = No fault, 1 = Load fault detected)
12	Programmer Segment Synchronise (0 = Waiting, 1 = Running)	New Alarm
13	Remote input sensor break (0 = Good, 1 = Bad)	Remote input sensor break (0 = Good, 1 = Bad)

Example: Fast read the status byte from a Series 2000 instrument at device address 02.

Command:

Device address	Function code	CRC	
02	07	41	12

Reply:

Device address	Function code	Fast readstatus byte	CRC	
02	07	30	D2	24

In this example the value of status byte (30h) has the following information;
PV is in sensor break
Instrument is in Manual mode

3.18 DIAGNOSTIC LOOPBACK

Function code: 08, (08h)

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

Command:

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

Reply:

The reply to function 08 is the same as the command T

Example: Perform a loopback from the Series 2000 instrument at address 2, using a data value of 1234h.

Command:

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

Reply:

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

3.19 WRITE N WORDS

Function code: 16, (10h)

Command:

Device address	Function code10	Address of first word		Number of words to write		Number of data bytes (n)	Data	CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	1 byte	n bytes	MSB	LSB

The maximum number of words that can be transmitted is

Series 2200: 32
Series 2400: 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.

A device address 00 will broadcast the data to all devices on the network.

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 downloading ramp/dwell programs, recipes, or 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. See also Chapter 6 - 'ignoring Modbus errors'.*

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: There will be no reply to a command broadcast to the device address 00. See the section on 'Error Response' below for details of the reply if the operation fails.

Device address	Function code 10	Address of first word		Number of words written		CRC	
1 byte	1 byte	MSB	LSB	MSB	LSB	MSB	LSB

Example: Write to the Series 2000 slave at device address 2 which is configured with full resolution.

Setpoint 3 = 12.3 (123) parameter address 164
Setpoint 4 = 15.0 (150) parameter address 165
Setpoint 5 = 25.0 (250) parameter address 166

Command:

Device address	Function code	Address of first word		Number of words to write		Number of data bytes	Data	CRC	
02	10	00	A4	00	03	06	See below	20	71

Data (123) for address 164		Data (150) for address 165		Data (250) for address 166	
01	7B	03	96	00	FA

Reply:

Device address	Function code	Address of first word		Number of words written		CRC	
02	10	00	A4	00	03	C1	D8

3.20 ERROR RESPONSE

The JBUS and MODBUS 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.)

The error response code indicates the type of error detected.
Series 2000 instruments support the following error response codes:

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

3.21 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.

A wait period is also required after a broadcast communication to device address 0.
Caution: Failure to observe the wait period after a broadcast will negate the broadcast message.

The wait period should exceed the instrument latency plus the message transmission time. Typical wait periods, for a single parameter read, are 20ms for 2400 and 50 to 100ms for 2200.

3.22 LATENCY

The time taken for the Series 2000 instrument 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), write 1 bit (function 05h), fast read of status (function 07h), and loopback (function 08h) are processed within a latency of between 2 and 10ms.

For the parameter functions, read n bits (function 01h), read n words (function 03h), and write n words (function 10h) the latency is indeterminate. The latency will depend on the instrument activity and the number of parameters being transferred and will take from 2 to 500ms, for 2400, and 50 to 500ms, for 2200.

It is possible to artificially increase the latency by setting the 'Comms Delay' parameter in the Mod HA 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.

3.23 MESSAGE TRANSMISSION TIME

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

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

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).

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 22.5 ms, (6 + 6.5 + 10.0).

For a broadcast command, (device address 0), the master would not expect a reply. In this case the wait period will exceed 16 ms, (6 +10.0).

4. CHAPTER 4 EI-BISYNCH PROTOCOL

This chapter introduces the principles of the EI-Bisynch communication protocol. EI-Bisynch is a proprietary Eurotherm protocol based on the ANSI X3.28-2.5 A4 standard for message framing. Despite its name, it is an ASCII based asynchronous protocol. Data is transferred using 7 data bits, even parity, 1 stop bit.

4.1 EXPLANATION OF TERMS

4.1.1 Address

Each instrument has a configurable address consisting of two digits, the fist being a 'group' number 0 to 9, and the second a 'unit number 0 to 9. There are, in principle, therefore 100 different addresses that may be used, 00 to 99, although in Series 2000 instruments, address 00 is reserved for use in configuration mode, leaving addresses 01 to 99 available.

The address is set on the CmS LiSt, using the Addr parameter. It may be necessary to use FuLL user interface via the ACCS LiSt to view and change the value of this parameter; see the instrument manual for more information.

4.1.2 Mnemonics

EI-Bisynch identifies parameters within an instrument using what are known as 'mnemonics'. These are usually two letter abbreiviations for a given parameter, for example PV for Process Variable, OP for Output, SP for Setpoint, and so on. Tables giving the mnemonics for parameters used in the 2000 Series is provided in Chapter 5.

4.1.3 Channels

EI-Bisynch provides for 'channel' data. This would be used, for example, when a single physical unit contains several independent control loops, each having their own Process Variable, Setpoint, and Output Power. In this case, the values for each loop are obtained by specifying different channel numbers: '1', '2', etc.

The 2000 series supports an optional channel number of '1', since it is a single loop controller. Other channel numbers will be rejected as invalid, with the exception of channel '9' which has a specialised function described elsewhere in this manual.

The channel number, if used, is encoded as a single ASCII character preceding the mnemonic, for example 1PV.

4.1.4 ASCII codes

Before a character is transmitted it is turned into an ASCII code. This is a universal code and a full list is included in Appendix B.

The ASCII code is 7 bits and to this the start, parity and stop bits have to be added as shown in the attached diagram. Eurotherm's protocol requires even parity and a single stop bit. This has to be programmed as part of the software concerning the communications output port of the computer.

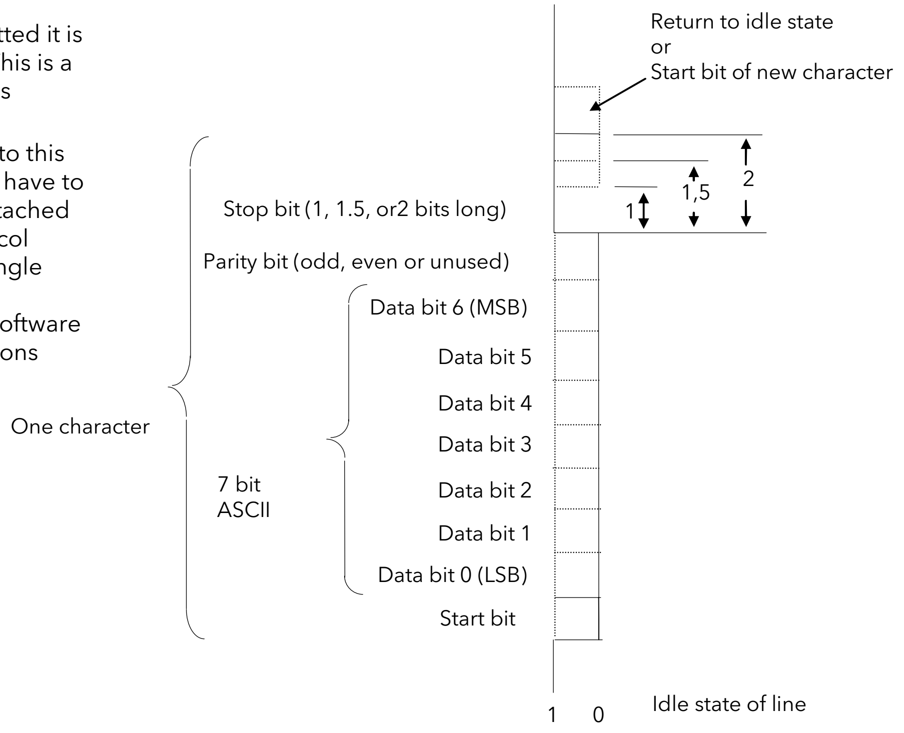


Figure 4-1: Asynchronous ASCII Character Format

4.2 CONTROL CHARACTERS

Several ASCII control characters are used in the framing of EI-Bisynch messages. These are:

Hex Value	Name	Usage
02	STX	Start of data in a message
03	ETX	End of message
04	EOT	End of transmission sequence
05	ENQ	Enquiry for a value
06	ACK	Positive Acknowledge
15	NAK	Negative Acknowledge

4.3 DATA FORMATS

Data in Bisynch messages is sent as a sequence of ASCII printable characters. Two principal data formats are used in Series 2000 instruments, Free format, and hex format. See also Chapter 6.

4.4 FREE DATA FORMAT

Parameter values returned from the instrument in 'free format' are of variable length. The instrument returns the value as it would be displayed on its front panel, with no leading or trailing spaces, e.g:

-99.9
123.4
123 (*integer value*)

Note that trailing decimal point characters are suppressed. Any 'sign' must precede the number itself.

Values written to the instrument may contain leading and trailing spaces, leading or trailing zeros, or sign indications.

This format is used for almost all parameters available over EI-Bisync in the 2000 series, with the exception of a few status words and prime set parameters which use 'hex format'.

NB: Because the returned value is of variable length, it is necessary to use the terminating ETX character to delimit the data value. It is not usually possible to make assumptions regarding the number of characters used to represent a value unless you are operating in a very restricted numeric range, 10.0 to 99.9 for example, where all possible values will take 4 characters to transmit.

4.5 HEX DATA FORMAT

This format is used for a few status words and prime set parameters. The value is preceded by a '>' (hex 3E) character, and normally consists of 4 hexadecimal characters, although it is acceptable to suppress leading zeroes when parameters are written. These characters represent the value of a 16 bit unsigned integer in hexadecimal (base 16) notation. Upper or lower case representations of 'A' to 'F' are acceptable, although the instrument will always return upper case. For example

>2040	equivalent to 8256 decimal
>ABCD	equivalent to 43981 decimal

This format is also used in conjunction with the channel 9 specifier to set up instrument scroll lists. More information on this operation is given later in this manual.

4.6 READING DATA FROM THE 2000 SERIES

To read data, a 'poll' message is issued to the instrument. This message takes the following format:

[EOT](GID)(GID)(UID)(UID)(CHAN)(C1)(C2)**[ENQ]**

Each item in the above description represents a single ASCII character. The items in bold type and square brackets are control characters used to 'frame' the message, whose values may be determined by reference to the table on P4.2. The bracketed items in normal type have the following significance:

GID The group id, or the first digit of the instrument address. E.G. '1' (31 hex) for instrument address 12, '0' (30 hex) for instrument address 1 (which is equivalent to address 01). The GID is sent twice, as a validation mechanism,

UID The unit id, or the second digit of the instrument address. E.G. '2' (32 hex) for instrument address 12, '1' (31 hex) for instrument address 1. The UID is sent twice, as a validation mechanism.

CHAN The channel number, which is optional. If used, send a value of '1' for Series 2000 single loop controllers.

C1 The first character of the mnemonic for the parameter being accessed, e.g. 'P' for Process Variable.

C2 The second character of the mnemonic for the parameter being accessed, e.g. 'V' for Process Variable.

If the instrument receives the message correctly and the mnemonic is valid, it will reply with

[STX](CHAN)(C1)(C2)<DATA>**[ETX]**(BCC)

CHAN Echo of the channel number from the poll message, if used. Otherwise not returned

C1, C2 Echo of the mnemonic from the poll message.

DATA The value of the parameter in a given display format. E.G 99.9, 1.2, -999, >1234 etc.

BCC This is a block checksum that is generated for data validation. It is computed by XORing (exclusive or) all the characters after and **excluding** the STX, and **including** the ETX. Note that it may take the value of 'EOT' and care must be taken when writing a protocol driver to ensure that this is not seen as an 'End of Transmission' sequence.

If a request is made for a non existent mnemonic, or a mnemonic representing a parameter that is not configured, the instrument will reply with a single 'EOT' character. If there is no reply at all, one of the following errors is possible:

- Incorrect wiring or faulty hardware (Cable, PC, RS422/485 adaptor, Instrument Comms Module)
- Instrument Address set wrong (PC, Instrument)
- Wrong Line set-up, should be 7 data bits, even parity, 1 stop (PC)
- Baud rate set wrongly (PC, Instrument)
- Parity error detected by instrument (Suspect line noise)
- Incorrect message format (PC)

To determine the cause of any communications problems, work systematically through the possible causes.

Example of a Parameter Read

For example, when reading PV from instrument address 1, the following sequence of characters will be sent and received:

Master: **[EOT]**0011PV**[ENQ]**
Instrument: **[STX]**PV16.4**[ETX]{BCC}**

Note that the BCC is a single character, that in this case has a value of 18 hex.

In hexadecimal, the transaction is as follows:

Master: 04 30 30 31 31 50 56 05
Instrument: 02 50 56 31 36 2E 34 03 18

4.7 FAST POLL

Fast polling provides a means of rapidly obtaining a set of parameter values, following an initial successful parameter read carried out as specified above. To read the next parameter in the 'fast poll' list, the master must send an 'ACK' control character following receipt of a valid poll response message, whereupon a reply message in the same format as the poll response will be received.

Parameters are only returned if they are configured, and the order in which parameters are returned sorted alphabetically by mnemonic, excluding program segment data. The only real use of this facility is to build an image of the parameter database in an instrument, and its use in other scenarios is not recommended.

4.8 REPEATED POLLING OF THE SAME PARAMETER

This facility provides a means of repeatedly obtaining a particular parameter value, following an initial successful parameter read carried out as specified above. To repeat the last poll operation, the master should transmit a 'NAK' control character following receipt of a valid poll response message, whereupon a reply message in the same format as the poll response will be received.

4.9 WRITING DATA TO THE 2000 SERIES

To write data, a 'select' message is issued to the instrument. This message takes the following format:

```
[EOT](GID)(GID)(UID)(UID)[STX](CHAN)(C1)(C2)<DATA>[ETX](BCC)
```

Each item in the above description represents a single ASCII character. The items in bold type and square brackets are control characters used to 'frame' the message, whose values may be determined by reference to the table on P 4.2. The bracketed items in normal type have the following significance:

GID The group id, or the first digit of the instrument address. E.G. '1' (31 hex) for instrument address 12, '0' (30 hex) for instrument address 1 (which is equivalent to address 01). The GID is sent twice, as a validation mechanism,

UID The unit id, or the second digit of the instrument address. E.G. '2' (32 hex) for instrument address 12, '1' (31 hex) for instrument address 1. The UID is sent twice, as a validation mechanism.

CHAN The channel number, which is optional. If used, send a value of '1' for Series 2000 single loop controllers.

C1 The first character of the mnemonic for the parameter being accessed, e.g. 'P' for Process Variable.

C2 The second character of the mnemonic for the parameter being accessed, e.g. 'V' for Process Variable.

DATA The value of the parameter in a given display format. E.G 99.9, 1.2, -999, >1234 etc.

BCC This is a block checksum that is generated for data validation. It is computed by XORing (exclusive or) all the characters after and **excluding** the STX, and **including** the ETX.

If a parity or address format error is detected, the instrument will not reply. Otherwise, the instrument will reply with either:

[NAK] Failed to write: BCC is incorrect, or Parameter not available or not configured, or Parameter is read only, or attempt has been made to read a parameter that is outside limits. A read of the EE mnemonic will give more information.

OR

[ACK] Parameter write was successful.

If there is no reply at all to a write request, one of the following errors is possible:

- Incorrect wiring or faulty hardware (Cable, PC, RS422/485 adaptor, Instrument Comms Module)
- Instrument Address set wrong (PC, Instrument)
- Wrong Line setup, should be 7 data bits, even parity, 1 stop (PC)
- Baud rate set wrongly (PC, Instrument)
- Parity error detected by instrument (Suspect line noise)
- Incorrect message format (PC)

To determine the cause of any communications problems, work systematically through the possible causes.

Example of a Parameter Write

For example, when writing a value of 22.0 to the setpoint to an instrument at address 1, the following sequence of characters will be sent and received:

```
Master:       [EOT]0011[STX]SL22.0[ETX]{BCC}
Instrument:                                  [ACK]
```

Note that the BCC is a single character, that in this case has a value of 32 hex.

In hexadecimal, the transaction is as follows:

```
Master:       04 30 30 31 31 02 53 4C 32 32 2E 30 03 02
Instrument:                                             06
```

4.10 BROADCASTS

A particular GID (group identifier) and UID (unit identifier) address is reserved for use in broadcast messages; the tilde character '~', hex 7E, can be used as a wild card in the GID and/or UID to selectively broadcast to all or to specific groups of instruments. In this case, no reply to the write request will be sent by any instrument. The only foolproof method of determining whether the write has succeeded is therefore to read back the parameter from each instrument and verify that it is set correctly. ***Broadcast is only recommended for systems where failure to communicate a value due to physical failure of wiring or other equipment will not cause damage to equipment.***

4.11 ERROR CODES HELD IN EE

The special 'EE' mnemonic may be used to give the status of the last communications transaction. It is a hex format parameter, and the values it may contain are as follows:

0:	No Error
1:	Invalid Mnemonic
2:	Parameter is read only
7:	Incorrect message
8:	Limit error

BISYNCH MESSAGE LATENCY

The time taken for the Series 2000 instrument 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.

Parameter read functions for 2000 Series EI-Bisynch take place with a latency of between 2 and 10ms.

Parameter write functions for 2000 Series EI-Bisynch take place with a latency of between 5 and 50ms.

It is possible to artificially increase the latency by setting the 'Comms Delay' parameter in the Mod HA 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.

4.12 MESSAGE TRANSMISSION TIME

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

$$\text{Message transmission time} = \frac{\text{Number of bytes in the message* Number of bits per character}}{\text{Baud rate}}$$

The number of bytes in a message can be determined by reference to the information given above.

The number of bits per character will be ten. (1 start bit, 7 data bits, an even parity bit and 1 stop bit).

5. CHAPTER 5 MODBUS AND EI- BISYNCH ADDRESSES

This section of the manual provides a list of all parameters in Series 2000 controllers that are available over the communications link. As far as possible, it follows the same organisation as the controller user interface itself. Definitions of parameters and status information not available via the controller display are also provided.

2000 Series controllers may be configured for a wide variety of functions and some parameters will only be available if the related function is configured. Modbus addresses that are not supported have no parameter assigned. In normal operating mode all configuration parameters are read only. To be able to write to these parameters, the controller must be in configuration mode.

If the Modbus protocol is used to read a parameter that is not configured, an undefined value will be returned. The EI-Bisynch protocol will return an EOT character to signal that the parameter is not available.

Modbus function 6 single parameter write operations to unconfigured or read only parameters will be rejected with a Modbus 'data error' return code. The EI-Bisynch protocol rejects such requests with the NAK character.

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 ramp/dwell program downloading, recipes, or 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 controller 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.

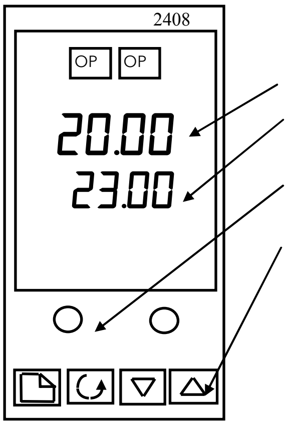
Rules for read and write operation in the Modbus IEEE area are dealt with in Chapter 3.

5.1 MODBUS AND BISYNCH ADDRESSES.

This section of the manual provides a list of all parameters in Series 2000 controllers that are available over the communications link. As far as possible, it follows the same organisation as the instrument user interface itself. Definitions of parameters and status information not available via the instrument display are also provided.

5.2 OPERATING MODE PARAMETERS

It is often only necessary to access a limited number of the most common parameters, where, for example, it is required to emulate the front panel of a controller in a mimic diagram. The following table shows a summary of common parameters:



Example 1 PID Controller		
Parameter	Modbus Address	Bisynch Address
Read Process value	1	PV
Change Setpoint	2 - (enter new value)	SL - (enter new value)
Raise Setpoint	2 - (new value in repeated steps)	SL - (new value in repeated steps)
Select Manual Mode	273 - (enumerator 1)	mA - (enumerator 1)
Change Output Power	3 - (new value)	OP - (new value)
Raise Output Power	3 - (new value in repeated steps)	OP - (new value in repeated steps)
Read Output Power	3	OP

Example 2 Valve Positioner		
Parameter	Modbus address	Bisynch Address
To Select Manual	273 - (enumerator 1)	mA - (enumerator 1)
To Change Output Position	60 - (new value)	VM - (new value)
To Read Output Position	53	VP

5.3 MODBUS AND BISYNCH PARAMETER TABLES

Notes: The following notes apply throughout this section

1. Issued software versions to date are 2400: 1.03, 2.04, 3.04 and 3.05 and 2200: 1.00, 1.20, 1.30 and 2.10.
2. Bi-synch only available in 2200 software versions 2 onwards
3. Greyed out cells indicate parameter not available
4. In Bisynch Lists, the numerals "zero" and "one" are shown in italics as *0* and *1.* This is to distinguish between I (as in In), l (as in low) and O (as in Out)

	Home list	**2400 Version 3**			**2200 Version 2**		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
	Process Variable	1	PV		1	PV	
OP	% Output level	3	OP		3	OP	
uPoS	Valve position				53	VP	
SP	Target setpoint (if in Manual mode)	2	SL		2	SL	
m-A	Auto-man select 0: Auto 1: Manual	273	mA		273	mA	
AmPS	Heater current (With PDSIO mode 2)	80	LI		80	LI	
CJ d	Customer defined identification number	629	ID		629	ID	
w.SP	Working set point. Read only: use Target set point or currently selected set point (1 to 16) to change the value	5	SP		5	SP	
OP See Note 1 below	Control output (on/off controller). Not writable unless the controller is in 'manual' mode. 0: -100% 1: 0% 2: 100%	85	OO	See Note 1 below			
-	VP Manual Output (alterable in Man only)	60	VM				
-	Valve Posn (computed by VP algorithm)	53	VP				
dı SP	Display 0: Standard 1: Load current 2: Output power 3: Program state 5: Blank 6: Valve position				106	WC	

Note 1:- In 2400 controllers S/W versions 3.00 onwards this parameter has been replaced by % output level.

OP	Control output (on/off) controller The power levels must be written as follows: Cool -100% OFF 0% Heat 100%	Modbus	Bisynch
		3	OP

run	Run List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
PrG	Current program running (active prog no.)	22	PN				
StAt	Program Status 1: Reset 2: Run 4: Hold 8: Holdback 16: Complete	23	PC				
PSP	Programmer setpoint	163	PS				
CYC	Program cycles remaining	59	CL				
SEG	Current segment number	56	SN				
StYP	Current segment type 0: End 1: Ramp (Rate) 2: Ramp (Time to target) 3: Dwell 4: Step 5: Call	29	CS				
SEGt	Segment time remaining	36	TS				
tGt	Target setpoint (current segment)	160	CT				
rAtE	Ramp rate	161	CR				
PrGt	Program time remaining	58	TP				
FASt	Fast run 0: No 1: Yes	57	FR				
out.1	Logic 1 output (current program) 0: Off 1: On (applies to all 8 logic outputs)	464	z1				
out.2	Logic 2 output (current program)	465	z2				
out.3	Logic 3 output (current program)	466	z3				
out.4	Logic 4 output (current program)	467	z4				
out.5	Logic 5 output (current program)	468	z5				
out.6	Logic 6 output (current program)	469	z6				
out.7	Logic 7 output (current program)	470	z7				
out.8	Logic 8 output (current program)	471	z8				
SYnc	Segment synchronisation 0: No 1: Yes	488	ut				
SEG.d	Flash active segment in lower display	284	fs	not in v1 & v2			

AL	Alarm List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
1---	Alarm 1setpoint value	13	A1		13	A1	
2---	Alarm 2setpoint value	14	A2		14	A2	
3---	Alarm 3setpoint value	81	A3		81	A3	
4---	Alarm 4setpoint value	82	A4		82	A4	
HY 1	Alarm 1 hysteresis	47	n5		580	HA	
HY2	Alarm 2 hysteresis	68	n6		580	HA	
HY3	Alarm 3 hysteresis	69	n7		580	HA	
HY4	Alarm 4 hysteresis	71	n8		580	HA	
Lbt	Loop break time 0: Off	83	lt		83	lt	
di AG	Enable diagnostic messages 0: No Diagnostics 1: Diagnostics	282	DM				

Atun	Autotune List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
tunE	Autotune enable 0: No Tune 1: Tune	270	AT		270	AT	
drA	Adaptive tune enable 0: No Adaotive Tune 1: Tune	271	AA				
drA.t	Adaptive tune trigger level	100	TR				
Adc	Automatic droop compensation (manual reset) 0: Manual reset 1: Calculated	272	DT		272	DT	

Pid	PID List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
G.SP	Gain scheduler setpoint	153	GS				
SEt	Current PID set (read only if gain scheduling is selected) 0: Set 1 1: Set 2	72	Gn				
Pb	Proportional band PID1	6	XP		6	XP	
ti	Integral time PID1 0: Off	8	TI		8	TI	
td	Derivative time PID1 0: Off	9	TD		9	TD	
rES	Manual reset PID1	28	MR		28	MR	
Hcb	Cutback high PID1 0: Auto	18	HB		18	HB	
Lcb	Cutback low PID1 0: Auto	17	LB		17	LB	
rEL.c	Relative cool gain PID1	19	RG		19	RG	
Pb2	Proportional band PID2	48	P2				
ti2	Integral time PID2 0: Off	49	I2				
td2	Derivative time PID2 0: Off	51	D2				
rES.2	Manual reset PID2	50	M2				
Hcb2	Cutback high PID2 0: Auto	118	hb				
Lcb2	Cutback low PID2 0: Auto	117	lb				
rEL2	Relative cool gain PID2	52	G2				
Pb.c	Cool (Brabender) proportional band	90	CP				
db.c	Cool (Brabender) deadband	91	CD				
FF.Pb	Feedforward proportional band	97	FP				
FF.tr	Feedforward trim	98	FO				
FF.du	Feedforward trim limit	99	FD				

On.OF	On/Off List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
hYS.H	Heat hysteresis	86	HH	These parameters appear in the output list in 2400 series	86	HH	
hYS.C	Cool hysteresis	88	hc		88	hc	
HC.db	Heat/cool deadband	16	HC		16	HC	
Sb.OP	On/Off sensor break output power 0: -100% 1: 0% 2: 100%	40	BO				

mtr	Motor List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
tm	Valve travel time	21	TT				
I n.t	Valve inertia time	123	vI				
bAc.t	Valve backlash time	124	Vb				
mP.t	Minimum pulse time	54	VT				
u.br	Bounded sensor break strategy	128	VS				
Sb.oP	VP Bounded sensor break	62	VB	VP b (feedback) controllers only			

NOTE; The following motor list is applicable to earlier (versions 1 & 2) valve position controllers only

mtr	Motor List	2400 versions1 and 2			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
cYc.t	VP Cycle time	132	vC				
I nu	VP Raise inertia 0: Off	123	vI				
I nd	VP Lower inertia 0: Off	130	vi				
bAc.u	VP Raise backlash 0: Off	124	vB				
bAc.d	VP Lower backlash 0: Off	129	vb				
UEL.u	VP Raise velocity limit	125	VR				
UEL.d	VP lower velocity limit	126	VL				
Pot.L	VP Position low limit	42	Le				
Pot.H	VP Position high limit	43	LE				
Sb.OP	Boundless sensor break o/p 0: Rest 1: Up 2: Down	128	VS				

SP	Setpoint list	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
SSEL	Select setpoint 0: SP1 1: SP2	15	SS	SP1 & SP2 available in standard controller	15	SS	SP1 & SP2 available in standard controller
	2: SP 3 3: SP 4 4: SP 5 5: SP 6 6: SP 7 7: SP 8 8: SP 9 9: SP 10 10: SP 11 11: SP 12 12: SP13 13: SP14 14: SP15 15: SP16			SP1 to SP16 available to order in the 16 setpoint option			
L-r	Local or remote setpoint select 0: Local 1: Remote	276	rE		276	rE	
SP 1	Setpoint 1	24	S1		24	S1	
SP 2	Setpoint 2	25	S2		25	S2	
SP 3	Setpoint 3	164	S3				
SP 4	Setpoint 4	165	S4				
SP 5	Setpoint 5	166	S5				
SP 6	Setpoint 6	167	S6				
SP 7	Setpoint 7	168	S7				
SP 8	Setpoint 8	169	S8				
SP 9	Setpoint 9	170	S9				
SP 10	Setpoint 10	171	Sa				
SP 11	Setpoint 11	172	Sb				
SP 12	Setpoint 12	173	Sc				
SP 13	Setpoint 13	174	Sd				
SP 14	Setpoint 14	175	Se				
SP 15	Setpoint 15	176	Sf				
SP 16	Setpoint 16	177	Sg				
rm.SP	Remote setpoint	485	uq		26	R1	
rmt.t	Remote setpoint trim	486	ur				
rAt	Ratio setpoint	61	RS				
Loc.t	Local setpoint trim	27	LT		27	LT	
SP L	Setpoint 1 low limit	112	LS		112	LS	SP 1L
SP H	Setpoint 1 high limit	111	HS		111	HS	SP 1H
SP2.L	Setpoint 2 low limit	114	L2		114	L2	
SP2.H	Setpoint 2 high limit	113	H2		113	H2	
Loc.L	Local setpoint trim low limit	67	TL		67	TL	
Loc.H	Local setpoint trim high limit	66	TH		66	TH	
SPrr	Setpoint rate limit 0: Off	35	RR		35	RR	
Hb.tY	Holdback type for setpoint rate limit 0: Off 1: Low	70	rT				

SP	Setpoint list	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
	2: High 3: Band						
Hb	Holdback value for setpoint rate limit	65	rH				
The following parameters are applicable to software version 2, series 2200 programmer/controllers only.							
dwEL	Dwell segment 0: Off				62	DW	
End.t	Go to state at end of program 0: Dwell 1: Reset 2: Hold 3: Standby				517	pt	
ProG	Program state write 1: Reset 2: Run				57	pc	
StAt	Program state read 1: Off 2: Run 4: Hold 16 End 32: Dwell 64 Ramp				23	PC	

iP	Input List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
FiLt	Input 1 filter time constant 0: Off	101	Tc		101	Tc	
FiLt.2	Input 2 filter time constant 0: Off	103	TC				
F.1	Derived input function factor 1	292	F1				
F.2	Derived input function factor 2	293	F2				
The following two parameters only appear if 'Input Switching' is configured							
Hi.IP	Switchover transition region high	286	Ih	Input switching			
Lo.IP	Switchover transition region low	287	I1				
The following two parameters only appear if 'Custom Pyrometers' is configured							
Emi5	Emmisivity	38	PE	Custom pyrometers			
Emi5.2	Emmisivity input 2	104	E2				
PViP	Select input 1 or input 2	288	pv				
The following three parameters apply to software version 2 only							
CAL	User calibration enable 0: Factory 1: User	110	UC		110	UC	
CAL.5	Selected calibration point 0: None 1: Input 1 low 2: Input 1 high 3: Input 2 low 4: Input 2 high	102	Uc		108	AE	0: None 1: Adj low 2: Adj hi 3: N/A 4: N/A
CAL.L	Transducer Low Cal enable 0: No 1: Yes	109	Ae	Software version 2 only			
AdJ.L	Adjust low calibration point	145	L1				
CAL.H	Transducer High Cal enable 0: No 1: Yes	108	AE				
AdJ.H	Adjust high calibration point	144	Lh				
AdJ	User calibration adjust input 1	146	LC		63	AJ	
AdJ	User calibration adjust input 2	148	Lc				
OFS.1	Input 1 calibration offset	141	O1		127	OF	OFSt
OFS.2	Input 2 calibration offset	142	O2				
mV.1	Input 1 measured value	202	VA		202	VA	mV
mV.2	Input 2 measured value	208	VD				
CJC.1	Input 1 cold junction temp. Reading 2400	215	t5				
CJC	Input 1 cold junction temp. Reading 2200				215	t5	CJC
CJC.2	Input 2 cold junction temp. reading	216	t6				
Li.1	Input 1 linearised value	289	QY				
Li.2	Input 2 linearised value	290	QZ				
PV.SL	Currently selected setpoint	291	in				

oP	Output List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
OP.Lo	Low power limit	31	LO		31	LO	
OP.Hi	High power limit	30	HO		30	HO	
rOP.L	Remote low power limit	33	RC				
rOP.H	Remote high power limit	32	RH				
OPrr	Output rate limit 0: Off	37	OR				
FOP	Forced output level	84	FM				
CYC.H	Heat cycle time	10	CH		10	CH	
hYS.H	Heat hysteresis (on/off output)	86	HH				
ont.H	Heat output minimum on time 0: Auto	45	MH		45	MH	
CYC.C	Cool cycle time	20	C2		20	C2	
hYS.C	Cool hysteresis (on/off output)	88	hc				
ont.C	Cool output minimum on time 0: Auto	89	MC		89	MC	
HC.db	Heat/cool deadband (on/off output)	16	HC				
Sb.OP	Sensor break output power	34	BP		34	BP	
	Motor valve position - 2200 V2 only				21	TT	mtr

cmS	Comms. List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
Addr	Communications address	131	Ad		131	Ad	

iNFo	Information List	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
di SP	Configuration of lower readout display 0: Standard 1: Load current 2: Output power 3: Status 4: Program time 5: None 6: Valve position 7: Process value 2 8: Ratio setpoint 9: Selected program number 10: Remote setpoint	106	wc				
LoG.L	PV minimum	134	Sm				
LoG.H	PV maximum	133	SM				
LoG.A	PV mean value	135	SA				
LoG.t	Time PV above threshold level	139	St				
LoG.u	PV threshold for timer log	138	ST				
rES.L	Logging reset 0: Not reset 1: Reset	140	SR				
mCt	Processor utilisation factor	201	mt				
w.OP	Working output	4	WO				
SSr	PDSIO SSR status 0: Good 1: Load fail 2: Open 3: Heater fail 4: SSR fail 5: Sn fail	79	Ss				
FF.OP	Feedforward component of output	209	FN				
P OP	Proportional component of output	214	Xp				
I OP	Integral component of output	55	xI				
d OP	Derivative component of output	116	xD				
UEL	VP velocity signal	219	Vv				
uP S	VP motor calibration state 0: Start 1: Waiting 2: Open valve 3: BLUp/InDn 4: Ttup 5: Overshoot 6: InUp/BLDn 7: TT down 8: Open 9: Low lim 10: Stopping 11: Raise 12: Inert up 13: Lower 14: Low lim 15: Stopping 16: Lower 17: InDn/BL 99: Abort	210	vS				

5.4 MISCELLANEOUS STATUS AND COMMS-ONLY PARAMETERS

		2400 version 3			**2200 version 2**		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
	Remote input comms access parameter	26	RI				
	Process error	39	ER		39	ER	
	Setpoint rate limit holdback status 0: inactive 1: Active	41	rS				
	System error logged flag 0: No error 1: Error	73	EI				
	Ramp rate disable	78	RD				
	Slave controller target setpoint	92	st				
	Slave controller ramp rate	93	sr				
	Slave controller synch signal	94	ss				
	Remote SRL hold	95	sh				
	BCD input value	96	BI				
	Controller version number Format: >XXYY (hex) where XX is major version number, and YY is minor version number. Eg. >0304 corresponds to V3.04	107	V0 (HEX format)		107	V0 (HEX)	
	CNOMO Manufacturers identifier	121	-		121	-	
	Controller identifier in format >ABCD (hex), A = 2 (series 2000) B = Range number 2: 2200 4: 2400 C = Size 3: 1/32 din 6: 1/16 din 8: 1/8 din 4: ¼ din D = Type 0: PID/on-off 2: VP	122	II (HEX format)		122	II (HEX)	
	Bisynch comms status 0: No error 1: Invalid mnemonic 2: Parameter is read only 7: Incorrect message 8: Limit error	-	EE (Hex format)				
	DIN rail remote par	151	RP				
	VP low limit switch - open	120	vc				
	VP high limit switch- open	119	vo				
	VP motor calibrate enable 0: Off 1: On	46	vT				
	Instrument mode **NOTE: WRITING OTHER VALUES TO THIS PARAMETER MAY CAUSE DAMAGE TO CALIBRATION OR CONTROLLER CONFIGURATION!** 0: Normal 1: Standby	199	IM	The controller address changes to '00' when Instrument mode is changed to configuratio	199	IM	

		2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
	2: Configuration			n			
	PV millivolts from comms	203	t1		203	t1	
	Input test point enable	205	IE		205	IE	
	Sensor break sourced from Test	206	t3		206	t3	
	Filter initialisation flag	207	Fi		207	Fi	
	Maximum number of segments (8 or 16): Read only	211	ns				
	Edit program	-	EP				
	Freeze control flag 0: Controlling 1: Hold	257	FC				
	Sensor break status flag 0: Good 1: Sensor break	258	sb		258	sb	
	Power failed flag 0: Good 1: Power fail detected	259	PF				
	Loop break status flag 0: Good 1: Loop break	263	Lb				
	Integral hold status flag 0: Good 1: Integral hold	264	IH				
	Acknowledge all alarms 0: Good 1: Acknowledge all alarms	274	AK		274	AK	
	Setpoint rate limit active status 0: No setpoint rate limit 1: setpoint rate limit active	275	Ra				
	Setpoint rate limit complete status 0: Setpoint rate limit incomplete 1: Setpoint rate limit complete	277	Rc				
	Holdback disable 0: Holdback enabled 1: Holdback disabled	278	HD				
	Disable keys 0: Keys enabled 1: Keys disabled	279	DK		279	DK	
	Remote input status 0: Good 1: Fault	280	RF				
	Sync/Continue flag 0: Continue 1: Awaiting sync	281	SC				
	DC input remote fault 0: good 1: Fault	283	IF				
	Maximum input value in engineering units	548	QL				
	Minimum input value in engineering units	549	QM				
	Setpoint span	552	QN				

5.5 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. Examples are:

Alarm states

Auto/Manual selection

Remote/Local selection

Disable front panel keys etc.

Individual parameters exist for all status indicators that may be changed over the communications link, and these should be used for 'write operations'. The exception is the digital output telemetry status word, which may be written to, to set digital outputs, provided their function is configured to 'No Func'.

The 2200 series contains two Status Words

1. Summary Output Status Word
2. Control Status Word

These are both shown in the table below.

Note, the detailed differences in the bit definitions between 2200 & 2400 in the Summary Output Status Word.

Parameter		Modbus 2400	Bisynch 2400	Modbus 2200	Bisynch 2200	Display
Fast Status byte. Read Only (Also available via Modbus Function 7)		74	FS (HEX format)	74		-
BIT	DESCRIPTION					
Bit 0	Alarm 1 State (0 = Safe 1 = Alarm)			Alarm 1 State (0 = Safe 1 = Alarm)		
Bit 1	Alarm 2 State (0 = Safe 1 = Alarm)			Alarm 2 State (0 = Safe 1 = Alarm)		
Bit 2	Alarm 3 State (0 = Safe 1 = Alarm)			Alarm 3 State (0 = Safe 1 = Alarm)		
Bit 3	Alarm 4 State (0 = Safe 1 = Alarm)			Alarm 4 State (0 = Safe 1 = Alarm)		
Bit 4	Manual Mode (0 = Auto 1 = Manual)			Manual Mode (0 = Auto 1 = Manual)		
Bit 5	Sensor Break (0 = Good PV 1 = Sensor Broken)			Sensor Break (0 = Good PV 1 = Sensor Broken)		
Bit 6	Loop Break (0 = Good closed loop 1 = Open Loop)			Loop Break (0 = Good closed loop 1 = Open Loop)		
Bit 7	Heater Fail (0 = No Fault 1 = Load fault detected)			Heater Fail (0 = No Fault 1 = Load fault detected)		

Parameter	Modbus 2400	Bisynch 2400	Modbus 2200	Bisynch 2200	Display
Summary Output Status Word	75	SO	75	SO	-

BIT	DESCRIPTION	
0	Alarm 1 State (0 = Safe, 1 = Alarm)	Alarm 1 State (0 = Safe, 1 = Alarm)
1	Alarm 2 State (0 = Safe, 1 = Alarm)	Alarm 2 State (0 = Safe, 1 = Alarm)
2	Alarm 3 State (0 = Safe, 1 = Alarm)	Alarm 3 State (0 = Safe, 1 = Alarm)
3	Alarm 4 State (0 = Safe, 1 = Alarm)	Alarm 4 State (0 = Safe, 1 = Alarm)
4	Manual Mode (0 = Auto, 1 = Manual)	Manual Mode (0 = Auto, 1 = Manual)
5	Sensor Break (0 = Good PV, 1 = Sensor Broken)	Sensor Break (0 = Good PV, 1 = Sensor Broken)
6	Loop Break (0 = Good closed loop, 1 = Open Loop)	Loop Break (0 = Good Closed Loop, 1 = Open Loop)
7	Heater Fail (0 = No Fault, 1 = Load fault detected)	Heater Fail (0 = No Fault, 1 = Load Fault Detected)
8	Tune Active (0 = Auto Tune disabled, 1 = Auto Tune active)	Load Fail (0 = No Fault, 1 = Load Fault Detected)
9	Ramp/Program Complete (0 = Running/Reset, 1 = Complete)	Ramp/Program Complete (0 = Running/Reset, 1 = Complete)
10	PV out of range (0 = PV within table range, 1 = PV out of table range)	PV out of range (0 = PV within table range, 1 = PV out of table range)
11	DC control module fault (0= Good,. 1= BAD)	SSR Fail (0 = No fault, 1 = Load fault detected)
12	Programmer Segment Synchronize (0 = Waiting, 1 = Running)	New Alarm
13	Remote input sensor break (0 = Good, 1 = Bad)	Remote input sensor break (0 = Good, 1 = Bad)
14	IP1 Fault (PV Input)	Reserved
15	Reserved	Reserved

Parameter	Modbus 2400	Bisynch 2400	Modbus 2200	Bisynch 2200	Display
Control Status Word	76	CW (HEX format)	76	CW (HEX format)	-

BIT	DESCRIPTION	
0	Control algorithm Freeze	Control algorithm Freeze
1	PV input sensor broken	PV input sensor broken
2	PV out of sensor range	PV out of sensor range
3	Self Tune failed	Self Tune failed
4	PID servo signal	PID servo signal
5	PID debump signal	PID debump signal
6	Fault detected in closed loop behaviour (loop break)	Fault detected in closed loop behaviour (loop break)
7	Freezes the integral accumulator	Freezes the integral accumulator
8	Indicates that a tune has completed successfully	Indicates that a tune has completed successfully
9	Direct/reverse acting control	Direct/reverse acting control
10	Algorithm Initialisation flag	Algorithm Initialisation flag
11	PID demand has been limited.	PID demand has been limited.
12	Autotune enabled	
13	Adaptive tune enabled	Adaptive tune enabled
14	Automatic Droop compensation enabled	Automatic Droop compensation enabled
15	Manual / Auto mode switch	Manual / Auto mode switch

Parameter	Modbus 2400	Bisynch 2400	Modbus 2200	Bisynch 2200	Display
Instrument Status Word	77	IW (HEX format)			-

BIT	DESCRIPTION
0	Config/Oper mode switch
1	Disables limit checking
2	SRL ramp running (Read Only)
3	Remote setpoint active
4	Alarm acknowledge switch.
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved

Parameter	Modbus 2400	Bisynch 2400	Modbus 2200	Bisynch 2200	Display
Digital Input Status Word Note that the order of LA and LB is reversed relative to what might be expected.	87	DW (HEX format)			-

BIT	DESCRIPTION
0	H Interface module (0 = Off, 1 = On)
1	J Interface module (0 = Off, 1 = On)
2	1A module (0 = Off, 1 = On)
3	LB logic input (0 = Off, 1 = On)
4	LA logic input (0 = Off, 1 = On)
5	1B module telemetry (0 = Off, 1 = On)
6	1C module (0 = Off, 1 = On)
7	2A module (0 = Off, 1 = On)
8	2B module (0 = Off, 1 = On)
9	2C module (0 = Off, 1 = On)
10	3A module (0 = Off, 1 = On)
11	3B module (0 = Off, 1 = On)
12	3C module (0 = Off, 1 = On)
13	Reserved
14	Reserved
15	Reserved

Parameter		Modbus 2400	Bisynch 2400	Modbus 2200	Bisynch 2200	Display
Digital Output Telemetry Parameter Note that the order of LA and LB is reversed relative to what might be expected.		551	dt (HEX format)			-
BIT	DESCRIPTION					
0	H Interface module telemetry (0 = Off, 1 = On)					
1	J Interface module telemetry (0 = Off, 1 = On)					
2	1A module telemetry (0 = Off, 1 = On)					
3	LB logic telemetry (0 = Off, 1 = On)					
4	LA logic telemetry (0 = Off, 1 = On)					
5	1B module telemetry (0 = Off, 1 = On)					
6	1C module telemetry (0 = Off, 1 = On)					
7	2A module telemetry (0 = Off, 1 = On)					
8	2B module telemetry (0 = Off, 1 = On)					
9	2C module telemetry (0 = Off, 1 = On)					
10	3A module telemetry (0 = Off, 1 = On)					
11	3B module telemetry (0 = Off, 1 = On)					
12	3C module telemetry (0 = Off, 1 = On)					
13	AA relay telemetry (0 = Off, 1 = On)					
14	Reserved					
15	Reserved					

Parameter		Modbus 2400	Bisynch 2400	Modbus 2200	Bisynch 2200	Display
Program Logic Outputs		162	PO			-
BIT	DESCRIPTION					
0	Program Output 1 (0 = OFF 1 = ON)					
1	Program Output 2 (0 = OFF 1 = ON)					
2	Program Output 3 (0 = OFF 1 = ON)					
3	Program Output 4 (0 = OFF 1 = ON)					
4	Program Output 5 (0 = OFF 1 = ON)					
5	Program Output 6 (0 = OFF 1 = ON)					
6	Program Output 7 (0 = OFF 1 = ON)					
7	Program Output 8 (0 = OFF 1 = ON)					
8	Reserved					
9	Reserved					
10	Reserved					
11	Reserved					
12	Reserved					
13	Reserved					
14	Reserved					
15	Reserved					

5.6 MODBUS BIT ADDRESSABLE PARAMETERS

A few bit addressable parameters are provided to conform to the CNOMO Modbus standard, but in general status information should be obtained via the status words or single status parameters in the Modbus word address space.

Parameter	Modbus Bit (Coil) Address
Auto/Manual Mode 0: Auto 1: Manual	2
Alarm 1 Status 0: No Alarm 1: Alarm	5
Sensor Break Status 0: OK 1: Sensor Break	10

5.7 CONFIGURATION MODE PARAMETERS

To write parameters in this group, it is first necessary to set the instrument mode parameter (Bisynch 'IM', Modbus 199) 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. When using Bisynch, note also that the controller address changes to 0 while in configuration mode.

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 around 5 seconds. During this period it will not be possible to communicate with the controller.

NOTE: For 2200 series the Configuration Password is Bisynch 152, Modbus 'Pc'.

WARNING:
Be very careful not to write values other than 0 or2 to instrument mode, since this parameter is also used clear non-volatile memory and to perform various factory calibration procedures. Writing an incorrect value can, therefore, damage your controller.

InSt	Instrument Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
unit	Instrument unit 0: °C 1: °F 2: °K 3: None			See PV conf	516	Q1	
dEc.P	Decimal places in the displayed value 0: nnnn. 1: nnn.n 2: nn.nn			See PV conf	525	QD	
CtrL	Control type 0: PID 1: On/Off 2: Manual 3: VP (No feedback) 4: VP b (Feedback)	512	Q0	'Manual' does not appear in Control Type list	512	Q0	'Manual' does not appear in Control Type list. VP b not available
Act	Control action 0: Reverse 1: Direct	7	CA		7	CA	
CooL	Type of cooling 0: Linear 1: Oil 2: Water 3: Fan 4: Proportional to error 5: On/Off	524	Q9		524	Q9	4: N/A 5: N/A
ti.td	Integral and Derivative time units 0: Seconds 1: Minutes 2: Hours	529	QH				
dtYP	Derivative action on: 0: PV 1: Error	550	Qe				
m-A	Front panel Auto/Manual button 0: Enabled 1: Disabled	530	mk				
r-h	Front panel Run/Hold button 0: Enabled 1: Disabled	564	pk				

InSt	Instrument Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
PwrF	Power feedback enable 0: Off 1: On	565	Pe		565	Pe	
Fwd.t	Feed forward type 0: None 1: Power feedforward 2: Setpoint feedforward 3: PV feedforward	532	QO				
Pd.tr	Manual/Auto transfer PD control 0: No 1: Yes	555	QQ		555	QQ	0: Hold 1: Track
Sbr.t	Sensor break output 0: Sensor break (go to set value) 1: Hold (output)	553	QP		553	QP	
FOP	Forced manual output 0: No 1: Trac (returns to last value) 2: Step (steps to forced output level)	556	QR		556	QR	2: N/A
bcd	BCD input function 0: None 1: Select program number 2: Select SP number	522	BF				
GSch	Gain schedule enable 0: No (disabled) 1: Yes (enabled)	567	QW				

PV	Process Value Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
unIt	Instrument units 0: °C 1: °F 2: °K 3: None	516	Q1				See InSt ConF list
dEc.P	Decimal places in displayed value 0: nnnn 1: nnn.n 2: nn.nn	525	QD				See InSt ConF list
rnG.L	Low range limit	11	QC				See iP LiSt
rnG.H	High range limit	12	QB				

iP	Input Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
inPt	Input type 0: J Type 1: K Type 2: L Type 3: R Type 4: B Type 5: N Type 6: T Type 7: S Type 8: PL 2 9: Custom (factory) * 10: RTD * 11: Linear mV (+/- 100mV) 12: Linear V (0-10V) 13: Linear mA 14: Square root V 15: Square root mA 16: Custom mV 17: Custom V 18: Custom mA	12290	Q2		12290	Q2	0: J Type 1: K Type 2: L Type 3: R Type 4: B Type 5: N Type 6 T Type 7: S Type 8: PL 2 9: RTD * 10: Cust. * 11: Lin mV 12: Lin V 13: N/A 14: N/A 15: N/A 16: N/A 17: N/A 18: N/A
* Note change in order for the two parameters							
CJC	Cold junction compensation 0: Auto 1: 0°C 2: 45°C 3: 50°C 4: Off	12291	Q3		12291	Q3	4: N/A
ImP	Sensor break impedance 0: Off (disabled - linear inputs only) 1: Auto 2: Hi (> 5K) 3: Hi Hi (>15K)	12301	Q8		578	IP	
inP.L	Input value low	12307	Q5		12307	Q5	
inP.h	Input value high	12306	Q4		12306	Q4	
VAL.L	Displayed reading low	12303	Q7		12303	Q7	
VAL.H	Displayed reading high	12302	Q6		12302	Q6	
rnG.L	Low range limit			See PV	11	QC	
rnG.H	High range limit			List	12	QB	

The following parameters are only present if a custom curve has been factory downloaded							
in 1	Custom linearisation input 1	601	J1				
VAL.1	Display value corresponding to input 1	621	K1				
in 2	Custom linearisation input 2	602	J2				
VAL.2	Display value corresponding to input 2	622	K2				
in 3	Custom linearisation input 3	603	J3				
VAL.3	Display value corresponding to input 3	623	K3				
in 4	Custom linearisation input 4	604	J4				
VAL.4	Display value corresponding to input 4	624	K4				
in 5	Custom linearisation input 5	605	J5				
VAL.5	Display value corresponding to input 5	625	K5				
in 6	Custom linearisation input 6	606	J6				
VAL.6	Display value corresponding to input 6	626	K6				
in 7	Custom linearisation input 7	607	J7				
VAL.7	Display value corresponding to input 7	627	K7				
in 8	Custom linearisation input 8	608	J8				
VAL.8	Display value corresponding to input 8	628	K8				

SP	Setpoint Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
nSP	Number of setpoints	521	NS				
rmtr	Remote tracking 0: Off 1: Track	526	QE				
mtr	Manual tracking 0: Off 1: Track	527	QF				
Prtr	Programmer tracking 0: Off 1: Track	528	QG				
rmP.U	Setpoint rate limit units 0: /Sec 1: /Min 2: /Hour	531	QJ				
rmt	Remote setpoint configuration 0: None 1: Remote setpoint 2: Remote setpoint + local trim 4: Remote trim + local setpoint	535	QA				

AL	Alarm Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
AL 1	Alarm 1 type 0: Off 1: Full scale low 2: Full scale high 16: Deviation band 17: Deviation high 18: Deviation low 34: Load current low 35: Load current high 36: Input 2 full scale low 37: Input 2 full scale high 38: Working output low 39: Working output high 40: Working setpoint low 41: Working setpoint high	536	n1		536	n1	36: N/A 37: N/A 38: N/A 39: N/A 40: N/A 41: N/A
Ltch	Latching 0: No 1: Yes 2: Event 3: Manual reset	540	n9		540	n9	2: N/A 3: N/A
bLoc	Blocking 0: No 1: Yes	544	nd		544	nd	
AL 2	Alarm 2 type (types as alarm 1)	537	n2		537	n2	
Ltch	Latching (types as alarm 1)	541	na		541	na	
bLoc	Blocking (types as alarm 1)	545	ne		545	ne	
AL 3	Alarm 3 type (types as alarm 1)	538	n3		538	n3	
Ltch	Latching (types as alarm 1)	542	nb		542	nb	
bLoc	Blocking (types as alarm 1)	546	nf		546	nf	
AL 4	Alarm 4 type (types as alarm 1) plus 64: Rate of change	539	n4		539	n4	Rate of change not available in 2200 series
Ltch	Latching (types as alarm 1)	543	nc		543	nc	
bLoc	Blocking (types as alarm 1)	547	ng		547	ng	

PrOG	**Programmer Configuration**	**2400 version 3**			**2200 version 2**		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
PtYP	Programmer type 0: None 1: Single program 4: Four programs 20: Twenty programs	517	pt				
HbAc	Holdback 0: Applies to whole program 1: Applies to each segment	559	Hb				
Pwr.F	Power fail recovery 0: Ramp back 1: Reset 2: Continue	518	pp				
Srvo	Servo 0: Servo to PV 1: Servo to SP	520	ps				
out	Programmable event outputs Version 1 controllers: 0: None 3: Three 6: Six 8: Eight Versions 2 and 3 controllers: 0: None 1: Eight	558	NO				
SYNC	Synchronisation of programs 0: No 1: Yes	557	SY				

5.8 INPUT/OUTPUT MODULES

The following tables list all possible hardware module and fixed output identifiers. There are physical restrictions on the types of modules that may be fitted in particular slots, for example it is not possible to place an RS485 comms module in slot 1A. Refer to the relevant Installation and Operation Handbook for full details.

In general it is possible to perform writes to Module Identifier comms addresses if (and only if) there are no hardware modules fitted other than the communications adapter. This allows controllers to be configured in the absence of hardware modules.

LA	**Digital Input 1 Configuration**	**2400 version 3**			**2200 version 2**		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Identity 4: Logic	12352	k0		12352	k0	
Func	Input functions	12355	k3		12355	k3	
	192: None						192: None
	193: Manual mode select						193: Man
	194: Remote setpoint select						194: Rem
	195: Setpoint 2 select						195: SP 2
	196: PID set 2 select						
	197: Integral hold						197: Int Hld
	198: One-shot self tune enable						
	199: Adaptive tune enable						
	200: Acknowledge alarms						200: Ack
	201: Select full access level						
	202: Keylock						202: K/lock
	203: Up button						
	204: Down button						
	205: Scroll button						
	206: Page button						
	207: Run						
	208: Hold						
	209: Run/Hold						
	210: Reset						210: Reset
	211: Skip						
	212: Holdback enabled						
	213: Least significant BCD digit						
	214: 2nd digit						
	215: 3rd digit						
	216: 4th digit						
	217: 5th digit						
	218: Most significant digit						
	219: Setpoint rate limit enable						
	220: Prog. waits at end of segment						
	223: Run/Hold						
	224: Reset/Run						
	225: Standby						225: Stby
	226: PV select						
	227: Advance to end of segment						
	240: Amps						240: Amps

Lb	**Digital Input 2 Configuration**	**2400 version 3**			**2200 version 2**		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Identity 4: Logic	12416	l0		12416	l0	
Func	Input functions, as LA above	12419	l3		12419	l3	240: Not available
UAL.L	Low scalar	12431	lf				
UAL.H	High scalar	12430	le				

AA	**Alarm Relay Configuration (2400)** **Output 3 Configuration (2200)**	**2400 version 3**			**2200 version 2**		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module identity	12480	g0		12480	g0	1: Relay 2208/2204 only
Func	Module function 0: None 1: Digital 2: Heat (2208/04 only) 3: Cool (2208/04 only)	12483	g3		12483	g3	
di GF	For Func = di G the following appear in 2200 series controllers: 0: Alarm 1 1: Alarm 2 2: Alarm 3 3: Alarm 4 4: Manual 5: Sensor break 6: Loop break 7: Heater fail 8: Load fail 9: 10: PV out of range 11: SSR fail 12: 13: Remote fail				12486	g6	0: Alarm 1 1: Alarm 2 2: Alarm 3 3: Alarm 4 4: Manual 5: Sens break 6: Loop break 7: Htr fail 8: Load fail 9: Prog end 10: PV out rng 11: SSR fail 12: New alarm 13: Remote fail (2208/04 only)
SEnS	Sense of output 0: Normal 1: Inverted	12489	g9		12489	g9	
	If Func = di G the following appear Alarm 1 Alarm 2 Alarm 3 Alarm 4 Controller in manual Sensor break PV out of range Loop break Load failure Tuning in progress Voltage or mA output open circuit PDSIO module connection O/C New alarm End of program (or SP rate limit) Program synchronisation active Program event output active						
	Summary of AA configuration	12486	g6 (HEX)				
	Program summary OP AA configuration	12503	gn (HEX)				

HA	Comms Module Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module identity 0: None 7: Digital comms 8: PDSIO output	12544	h0		12544	h0	
Func	Module function For id = cmS 64: None 65: Modbus 66: Bisynch For id = PdS (Master) 128: None 129: PDSIO SP retransmission 130: PDSIO PV retransmission 131: PDSIO output power retrans. 133: PDSIO SP retrans, no holdback For id = PdSi (Slave) 96: None 97: PDSIO setpoint input	12547	h3		12547	h3	
bAud	Baud rate 0: 9600 1: 19200 2: 4800 3: 2400 4: 1200	12548	h4		12548	h4	
dELY	Delay. This introduces a short delay between messages to allow certain 'intelligent' RS485 converters to switch between RX and TX modes. 0: No - 0mS 1: Yes - 10mS	523	wt				
PrtY	Parity (Modbus only) 0: None 1: Even 2: Odd	12549	h5		12549	h5	
rES	Resolution (Modbus only) Changes are effective immediately 0: Full 1: Integer	12550	h6		12550	h6	rESn
VAL.L	Retransmitted value low	12559	hf		12559	hf	
VAL.H	Retransmitted value high	12558	he		12558	he	

JA	Comms Module 2 Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module identity 0: None 8: PDSIO output 9: PDSIO input	12608	j0				
Func	Module function For id = PdS 128: None 129: PDSIO SP retransmission 130: PDSIO PV retransmission 131: PDSIO output power retrans. 133: PDSIO SP retrans, no holdback For id = PdSi 96: None 97: PDSIO setpoint input	12611	j3				
UAL.L	Retransmitted value low	12623	jf				
UAL.H	Retransmitted value high	12622	je				

1A	Output 1A Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module identity 0: None 1: Relay output 2: DC output non-isolated 3: Logic/PDSIO output 4: Logic input 5: Triac output 10: Error/Bad module 11: DC retransmission 12: DC output isolated	12672	a0		12672	aO	0: None 1: Relay 2: DC out 3: Logic 5: Triac 10: Bad
Func	Module function For id = rELY LoG or SSr 0: None 1: Digital output 2: Heating output 3: Cooling output 4: Open motorised valve 10: PDSIO mode 1 heating 11: PDSIO mode 2 heating For id = dc.rE or dc.OP 16: None 17: Heating output 18: Cooling output 19: Retransmission of PV 20: Retransmission of SP 21: Retransmission of error 22: Retransmission of OP power For id = LoGi Use the enumerators in LA Config. list	12675	a3		12675	a3	0: None 1: Dig o/p 2: Heat 3: Cool Logic only 4: SSR1 5: SSR2 6: SSR3 DC output 16: None 17: Heat 18: Cool
diGF	For Func = diG the following appear in 2200 series controllers: 0: Alarm 1 1: Alarm 2 2: Alarm 3 3: Alarm 4 4: Manual 5: Sensor break 6: Loop break 7: Heater fail				12678	a6	As 2400 plus

1A	Output 1A Configuration	2400 version 3			2200 version 2		
	8: Load fail 10: PV out of range 11: SSR fail 13: Remote fail						9: Prog end 12: New alarm
VAL.L	% PID or Retran value giving min. o/p	12687	af				
VAL.H	% PID or Retran value giving max. o/p	12686	ae				
unit	Units 1: Volts 2: mA	12684	ac				
Out.L	Minimum electrical output	12689	ah		12689	ah	
Out.H	Maximum electrical output	12688	ag		12688	ag	
SEnS	Sense of output 0: Normal 1: Inverted	12681	a9		12681	a9	
	Summary output 1A configuration	12678	a6 (HEX)				
	DC output 1A telemetry parameter	12694	am				
	Program summary output 1A config	12695	an (HEX)				

1b	Output 1B Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module 1B identity	12673	a1				
Func	Module 1B function	12676	a4				
SEnS	Sense of output (nor/inv as1A)	12682	aa				
	Summary of 1B configuration	12679	a7 (HEX)				
	Summary program O/P 1B config.	12696	ao (HEX)				

1C	Output 1C Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module 1C identity	12674	a2				
Func	Module 1C function	12677	a5				
VAL.L	Module 1C value giving min output	12699	ar				
VAL.H	Module 1C value giving max output	12698	aq				
Out.L	Module 1C Minimum electrical output	12701	at				
Out.H	Module 1C Maximum electrical output	12700	as				
SEnS	Sense of output (nor/inv as 1A)	12683	ab				
	Summary of 1C configuration	12680	a8 (HEX)				
	Summary program O/P 1C config.	12697	ap (HEX)				

2A	Output 2A Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module identity 0: None 1: Relay output 2: DC output non-isolated 3: Logic/PDSIO output 4: Logic input 5: Triac output 10: Error/Bad module 11: DC retransmission 12: DC output isolated 13: Transmitter power supply 14: Potentiometer input (V position)	12736	b0		12736	b0	Only the following are relevant: 0: None 1: Relay 3: Logic 5: Triac 10: Bad
Func	Module function For id = rELY LoG or SSr 0: None 1: Digital output 2: Heating output 3: Cooling output 5: Close motorised valve For id = dc.rE or dc.OP 16: None 17: Heating output 18: Cooling output 19: Retransmission of PV 20: Retransmission of SP 21: Retransmission of error 22: Retransmission of OP power For id = Pot 160: None 161: Remote setpoint 162: Feedforward input 163: Remote OP power high 164: Remote OP power low 165: Valve position	12739	b3		12739	b3	Only the following are relevant: 0: None 1: Dig o/p 2: Heat 3: Cool 193: Man enab 194: Rem SP 195: 2nd SP 197: Int hold 200: Ack alms 202: Key lock 210: Reset prg 225: Standby
diGF	For Func = diG see 1A list for enumerators				12742	b6	
VAL.L	% PID or Retran value giving min. o/p	12751	bf				
VAL.L	Potentiometer input low scalar	12763	br				
VAL.H	% PID or Retran value giving max. o/p	12750	be				
VAL.H	Potentiometer input high scalar	12762	bq				
unit	Units 1: Volts 2: mA	12748	bc				
Out.L	Minimum electrical output	12753	bh				
Out.H	Maximum electrical output	12752	bg				
SEnS	Sense of output 0: Normal 1: Inverted	12745	b9		12745	b9	
	Summary output 2A configuration	12742	b6 (HEX)				
	DC output 2A telemetry parameter	12758	bm				
	Program summary output 2A config	12759	bn (HEX)				

2b	Output 2B Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module 2B identity	12737	b1				
Func	Module 2B function	12740	b4				
SEnS	Sense of output (nor/inv as 2A)	12746	ba				
	Summary of 2B configuration	12743	b7 (HEX)				
	Summary program O/P 2B config.	12760	bo (HEX)				

2C	Output 2C Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module 2C identity	12738	b2				
Func	Module 2C function	12741	b5				
SEnS	Sense of output (nor/inv as 2A)	12747	bb				
	Summary of 2C configuration	12744	b8 (HEX)				
	Summary program O/P 2C config.	12761	bp (HEX)				

3A	Output 3A Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module identity 0: None 1: Relay output 2: DC output non-isolated 3: Logic/PDSIO output 4: Logic input 5: Triac output 6: DC input 10: Error/Bad module 11: DC retransmission 12: DC output isolated 13: Transmitter power supply 14: Potentiometer input (V position)	12800	c0		12800	c0	0: None 1: Relay
Func	Module function For id = rELY LoG or SSr 0: None 1: Digital output 2: Heating output 3: Cooling output For id = dc.rE or dc.OP 16: None 17: Heating output 18: Cooling output 19: Retransmission of PV 20: Retransmission of SP 21: Retransmission of error 22: Retransmission of OP power For id = Pot 160: None 161: Remote setpoint 162: Feedforward input 163: Remote OP power high 164: Remote OP power low 165: Valve position For id = dC.iP 32: None 33: Remote setpoint 34: Feedforward input 35: Remote output power max. 36: Remote output power min. 37: PV = highest of ip1 or ip2 38: PV = lowest of ip1 or ip2 39: Derived function 40: Select ip1 or ip2 41: Transition of control - ip1 to ip2	12803	c3		12803	c3	0: None 1: Dig o/p 2: Heat 3: Cool
diGF	For Func = diG see 1A list for enumerators				12806	c6	
inPt	input type (input 2) Refer to input configuration for all types + HiIn	12830	cu				
CJC	Cold junction compensation (input 2) Refer to input configuration for types	12831	cv				
imP	Sensor break impedance (input 2) Refer to input configuration for types	12813	cz				
inP.L	Input value low	12819	cj				
inP.H	Input value high	12818	ci				
VAL.L	Input module 3A low value	12829	ct				
VAL.H	Input module 3A high value	12828	cs				

VAL.L	Module 3A low value	12815	cf				
VAL.L	Potentiometer input 3A low scalar	12827	cr				
VAL.H	Module 3A high value	12814	ce				
VAL.H	Potentiometer input 3A high scalar	12826	cq				
unit	Units 3A 1: Volts 2: mA	12812	cc				
Out.L	Minimum electrical output	12817	ch				
Out.H	Maximum electrical output	12816	cg				
SEnS	Sense of output 0: Normal 1: Inverted	12809	c9		12809	c9	
	Summary output 3A configuration	12806	c6 (HEX)				
	DC output 3A telemetry parameter	12822	cm				
	Program summary output 3A config	12823	cn (HEX)				

3b	Output 3B Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module 3B identity	12801	c1				
Func	Module 3B function	12804	c4				
SEnS	Sense of output (nor/inv as 3A)	12810	ca				
	Summary of 3B configuration	12807	c7 (HEX)				
	Summary program O/P 3B config.	12824	co (HEX)				

3C	Output 3C Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module 3C identity	12802	c2				
Func	Module 3C function	12805	c5				
SEnS	Sense of output (nor/inv as 3A)	12811	cb				
	Summary of 3C configuration	12808	c8 (HEX)				
	Summary program O/P 3C config.	12825	cp (HEX)				

4A	**Output 4A Configuration**	**2400 version 3**			**2200 version 2**		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
id	Module identity 0: None 1: Relay output	12864	e0	Not available in 2416	12864	e0	2204 only 0: None 1: Relay
Func	Module function 0: None 1: Digital output 2: Heating output 3: Cooling output	12867	e3	Not available in 2416	12867	e3	2204 only 0: None 1: Dig o/p 2: Heat 3: Cool
diGF	For Func = di G see 1A list for enumerators				12870		2204 only
VAL.L	Input module 4A low value	12879	ef	Not 2416			
VAL.H	Input module 4A high value	12878	ee	Not 2416			
Out.L	Minimum electrical output	12881	eh	Not 2416			
Out.H	Maximum electrical output	12880	eg	Not 2416			
SEnS	Sense of output (nor/inv as 3A)	12873	e9	Not 2416	12873	e9	2204 only
	Summary output 4A configuration	12870	e6 (HEX)	Not 2416			
	Program summary output 4A config	12887	en (HEX)	Not 2416			

CAL	**Calibration Configuration**	**2400 version 3**			**2200 version 2**		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
rcAL	Calibration node select 0: None 1: PV 1 2: PV 2 3: DC output high - module 1 4: DC output low - module 1 5: DC output high - module 2 6: DC output low - module 2 7: DC output high - module 3 8: DC output low - module 3	533	Cn				
PV	PV Calibration state 0: Idle 1: Select 0mV cal point 2: Select 50mV cal point 3: Select 0V cal point 4: Select 10V cal point 5: Select 0°C CJC cal point 6: Select 400 ohms cal point 7: Select 0V high impedance cal pt 8: Select 1V high impedance cal pt 9: Restore factory calibration 10: Busy	534	Ci				
GO	Start calibration No Yes Busy Done Fail	65535					
CAL.H	Module 1A output calibration high trim	12692	ak				
CAL.H	Module 2A output calibration high trim	12756	bk				

CAL.H	Module 3A output calibration high trim	12820	ck				
CAL.L	Module 1A output calibration low trim	12693	al				
CAL.L	Module 2A output calibration low trim	12757	bl				
CAL.L	Module 3A output calibration low trim	12821	cl				
UCAL	User calibration enable 0: No 1: Yes	566	te		566	te	AdJ
Pt 1.L	Low calibration point for input 1	563	QV		563	QV	Pnt.L
Pt 1.H	High calibration point for input 1	562	QU		562	QU	Pnt.H
OF 1.L	Offset low for input 1	561	QT		561	QT	OFS.L
OF 1.H	Offset high for input 1	560	QS		560	QS	OFS.H
Pt2.L	Low calibration point for input 2	571	Qd				
Pt2.H	High calibration point for input 2	570	Qc				
OF2.L	Offset low for input 2	569	Qb				
OF2.H	Offset high for input 2	568	Qa				

PASS	Password Configuration	2400 version 3			2200 version 2		
		Modbus	Bisynch	Notes	Modbus	Bisynch	Notes
ACC.P	Full or edit level password	514	QI		514	QI	
cnF.P	Configuration level password	515	QK		515	QK	

5.9 RAMP/DWELL PROGRAMMER DATA - MODBUS

This Section Applies To 2400 Series Controllers only

5.9.1 Program Data Organisation

A 2400 series controller can contain multiple "programs", each consisting of up to 16 segments. The data for each program starts at the base Modbus address given by the following table:

Program	Base Address (Decimal)	Base Address (Hex)
Program 0 (Currently Running Program - changes permitted only in hold, and are not permanently stored)	8192	2000
Program 1	8328	2088
Program 2	8464	2110
Program 3	8600	2198
Program 4	8736	2220
Program 5	8872	22A8
Program 6	9008	2330
Program 7	9144	23B8
Program 8	9280	2440
Program 9	9416	24C8
Program 10	9552	2550
Program 11	9688	25D8
Program 12	9824	2660
Program 13	9960	26E8
Program 14	10096	2770
Program 15	10232	27F8
Program 16	10368	2880
Program 17	10504	2908
Program 18	10640	2990
Program 19	10776	2A18
Program 20	10912	2AA0

The parameters used to describe a program are organised into 17 blocks, each of 8 words in length, starting at the base address for the program. There is one block for general program data, such as the units to be used for ramp and dwell times, and 16 further blocks for the segment data itself. To obtain the Modbus address of the data block for a given program, add the block offset given in the next table to the program

Contents	Offset (Decimal)	Offset (Hex)
Program General Data	0	0
Segment 1	8	8
Segment 2	16	10
Segment 3	24	18
Segment 4	32	20
Segment 5	40	28
Segment 6	48	30
Segment 7	56	38
Segment 8	64	40
Segment 9	72	48
Segment 10	80	50
Segment 11	88	58
Segment 12	96	60
Segment 13	104	68
Segment 14	112	70
Segment 15	120	78
Segment 16	128	80

5.9.2 Program General Data

The offsets of each parameter within the program general data block is given by the next table:

Address Offset	Parameter
0	HoldbackType 0: None 1: Low 2: High 3: Band
1	HoldbackValue
2	Ramp Units 0: Secs 1: Mins 2: Hours
3	Dwell Units 0: Secs 1: Mins 2: Hours
4	Program Cycles
5	Reserved
6	Reserved
7	Reserved

5.9.3 Program Segment Data

Program segment data is specified using 8 modbus addresses, with the contents varying depending on the type of the segment. The format per segment is detailed in the following table, which gives the offset from the start of a segment data block for each item.

Address Offset	Segment Types					
	STEP	DWELL	RAMP RATE	RAMP TIME TO TARGET	CALL	END
0	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
1	Target Setpoint		Target Setpoint	Target Setpoint		End Power
2		Duration	Rate	Duration		
3					Program Number	End Type
4	Logic O/P's	Logic O/P's	Logic O/P's	Logic O/P's	Call Cycles	Logic O/P's
5						
6						
7						

Example Address calculations

Program 1, Segment 4, Segment Type = 8328 + 32 + 0 = 8360 (20A8 Hex)
Program 2, Holdback Value = 8464 + 0 + 1 = 8465 (2111 Hex)
Program 4 Segment 16, End Type = 8872 + 128 + 3 = 9003 (232B Hex)

5.10 RAMP/DWELL PROGRAMMER DATA - EI-BISYNCH

The EP parameter is used to select the program in which the segment, or general data being accessed, refers to. For example, setting EP to 4 allows values in program number 4 to be read/written, (provided a '4 Programmer' controller is configured).

Mnemonics for accessing segment data and general program data are given in the table below.

Note: 'writes' to parameters that are not relevant for a given configuration or segment type will be rejected by the controller, which will reply with a NAK character.

Mnemonic	**Program general Data**
$0	Holdback Type
s0	Holdback Value
d0	Ramp Units
p0	Dwell Units
o0	Cycles

	Per Segment Data					
Mnemonic	**Step**	**Dwell**	**Ramp (Rate)**	**Ramp (Time to Tgt)**	**Call**	**End**
Segment 1						
$**1**	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s**1**	Target		Target	Target		
d**1**		Duration	Ramp Rate	Duration		
p**1**					Call Prog	End Type
o**1**	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 2						
$2	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s2	Target		Target	Target		
d2		Duration	Ramp Rate	Duration		
p2					Call Prog	End Type
o2	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 3						
$3	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s3	Target		Target	Target		
d3		Duration	Ramp Rate	Duration		
p3					Call Prog	End Type
o3	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 4						
$4	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s4	Target		Target	Target		
d4		Duration	Ramp Rate	Duration		
p4					Call Prog	End Type
o4	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 5						
$5	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s5	Target		Target	Target		
d5		Duration	Ramp Rate	Duration		
p5					Call Prog	End Type
o5	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 6						

	Per Segment Data					
Mnemonic	**Step**	**Dwell**	**Ramp (Rate)**	**Ramp (Time to Tgt)**	**Call**	**End**
$6	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s6	Target		Target	Target		
d6		Duration	Ramp Rate	Duration		
p6					Call Prog	End Type
o6	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 7						
$7	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s7	Target		Target	Target		
d7		Duration	Ramp Rate	Duration		
p7					Call Prog	End Type
o7	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 8						
$8	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s8	Target		Target	Target		
d8		Duration	Ramp Rate	Duration		
p8					Call Prog	End Type
o8	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals

	Per Segment Data					
Mnemonic	**Step**	**Dwell**	**Ramp (Rate)**	**Ramp (Time to Tgt)**	**Call**	**End**
Segment 9						
$9	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s9	Target		Target	Target		
d9		Duration	Ramp Rate	Duration		
p9					Call Prog	End Type
o9	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 10						
$:	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s:	Target		Target	Target		
d:		Duration	Ramp Rate	Duration		
p:					Call Prog	End Type
o:	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 11						
$;	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s;	Target		Target	Target		
d;		Duration	Ramp Rate	Duration		
p;					Call Prog	End Type
o;	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 12						
$<	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s<	Target		Target	Target		
d<		Duration	Ramp Rate	Duration		
p<					Call Prog	End Type
o<	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 13						
$=	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type

	Per Segment Data					
Mnemonic	**Step**	**Dwell**	**Ramp (Rate)**	**Ramp (Time to Tgt)**	**Call**	**End**
s=	Target		Target	Target		
d=		Duration	Ramp Rate	Duration		
p=					Call Prog	End Type
o=	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 14						
$>	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s>	Target		Target	Target		
d>		Duration	Ramp Rate	Duration		
p>					Call Prog	End Type
o>	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 15						
$?	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s?	Target		Target	Target		
d?		Duration	Ramp Rate	Duration		
p?					Call Prog	End Type
o?	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals
Segment 16						
$@	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type	Segment Type
s@	Target		Target	Target		
d@		Duration	Ramp Rate	Duration		
p@					Call Prog	End Type
o@	Digitals	Digitals	Digitals	Digitals	Digitals	Digitals

6. CHAPTER 6 ADVANCED TOPICS

6.1 ACCESS TO FULL RESOLUTION FLOATING POINT AND TIMING DATA (MODBUS ONLY)

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 Series 2000 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, notably those used for the programmer function are always returned over the communications link in seconds. 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 x 2 + 8000h = 8004h = 32772 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.

6.2 DATA TYPES USED IN SERIES 2000 INSTRUMENTS

- Enumerated parameters are parameters which have a textual representation for their value on the user interface, for example, 'Auto' or 'Manual', 'On' or 'Off', 'SP1', 'SP2', ...,'SP16', etc. A full list is included in the parameter tables in the previous chapter.

- Status words are generally only available over communications, and are used to group binary status information.

- 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.

- 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.

- Time Type parameters measure durations, and include Integral and Derivative times, program durations, etc.

6.3 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.

6.4 FLOATING POINT PARAMETERS

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

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

BIT 31	30	23	22	0
Sign	2^7	2^0	2^{-1} 2^{-2}	2^{-23}

{--- -------EXPONENT--------}{---------------------------FRACTION-----------------------------}

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

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

The format used to transfer the IEEE number is as follows

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

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

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

6.5 TIME TYPE PARAMETERS

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

6.6 USER INTERFACE ACCESS PERMISSIONS (MODBUS)

Some Series 2000 operating parameters may be hidden, made read only, or promoted to the 'main' scroll list. Additionally, certain parameter lists may be hidden. In Modbus, this operation may be performed by writing values to the address range 16384 to 32627.

To calculate the address used to set user interface permissions, take the normal Modbus address of the parameter involved, and add 16384 to it. List headers and 'special' user interface parameters are listed at the end of the parameter addresses in chapter 5 of this manual. You must be in configuration mode to write to the user interface access parameters, which use the following enumerations:

Parameters:

0	Hide Parameter
1	Promote Parameter to main scroll list
2	Parameter is read only
3	Display Parameter with default read/write status

List Headers

0	Hide List
3	Display List

6.7 USER INTERFACE ACCESS PERMISSIONS (EI-BISYNCH)

Some Series 2000 operating parameters may be hidden, made read only, or promoted to the 'main' scroll list. Additionally, certain parameter lists may be hidden. In Bisynch, this operation may be performed by using a channel number of 9 (e.g. 9TI, for Integral Time). List headers and 'special' user interface parameters are listed at the end of the parameter addresses in chapter 5 of this manual. You must be in configuration mode to write to the user interface access parameters, which use HEX format, and the following enumerations:

Parameters:

0	Hide Parameter
1	Promote Parameter to main scroll list
2	Make parameter read only
3	Display Parameter with default read/write status

List Headers

0	Hide List
3	Display List

6.8 PROGRAMMABLE LOGIC CONTROLLERS AND 2400 SERIES INSTRUMENTS

6.8.1 EI-Bisynch

This applies to instruments with a version number of 3.00 or greater only.

EI-Bisynch uses a variable length data field when sending or receiving numeric data. This means that the number of characters expected in response to a request for a parameter value is not known at the time of making the request. For example, a PV might be returned as 9.87 (4 characters, including the decimal point), 99.65 (5 characters), or even -99.99 (6 characters).

Many Programmable Logic Controllers use simple ‘Basic’ modules to provide serial comms, which often require that a certain, fixed, number of characters is expected in reply to a request made over serial comms. In order to permit use of Bisynch with such modules, it is possible to configure the instrument to return fixed field length data.The returned data contains leading ASCII space characters which make up the total returned field length to 8 characters. For example

```
‘    9.87’
‘   99.65’
‘  -99.99’
```

(Note that the quote (’) characters are not returned). This means that an example reply from an instrument for a read of PV would be as follows:

```
[STX]PV    9.87[ETX](BCC)
```

i.e. a total of 13 characters.

This does not apply to hex format, which always returns a field length of 5 characters (‘>ABCD’).

To switch this facility on, write `>0001` to the ‘FX’ mnemonic (hex format). The setting of this mnemonic is held in non-volatile memory and so you only need perform this operation once. To cancel the facility, write `>0000` to FX.

6.8.2 Modbus

There are many ways of connecting 2000 Series Instruments to Programmable Logic Controllers using Modbus, for example the ProSoft 3100/3150 MCM module for Allen Bradley PLC/5 and SLC/5. It is usually best to avoid the use of Basic modules which may result in very slow communications. Eurotherm 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 2000 Series supports standard Modbus RTU, allowing use of function 16 for block write operations, and functions 3 and 4 for reads.

Because Modbus modules often allow a restricted number of block operations, it is sometimes useful to create large blocks containing all the data to be written for a given instrument. Because the 2400 contains a mixture of read/write and read-only data, this can be difficult to achieve. Therefore, for 2400 firmware versions 3.00 and greater, a facility has been provided that allows block writes to continue even if values in the block are not currently writeable (the values that are not writeable are ignored, and there is no error return).

To switch this facility on, write a value of 1 to the instrument Modbus register 220. The setting of this register is held in non-volatile memory and so you only need perform this operation once. To cancel the facility, write 0 to register 220.

7. 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 RS232, RS422 and 485.
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.
RS422	This refers to the electrical standard used for signalling information on a serial communications link.
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

8. APPENDIX B. ASCII CODES

ASCII Codes	ASCII - HEX
STX - Start of Text	02
ETX - End of Text	03
EOT - End of Transmission	04
ENQ - Enquiry	05
ACK - Positive Acknowledge	06
NAK - Negative Acknowledge	15
Space	20
- Minus Sign	2D
. Decimal Point	2E
0	30
1	31
2	32
3	33
4	34
5	35
6	36
7	37
8	38
9	39
> (Greater Than)	3E

HEX-ASCII TABLE - complete list											
00	NUL	15	NAK	2B	+	40	@	56	V	6B	k
01	SOH	16	SYN	2C	,	41	A	57	W	6C	l
02	STX	17	ETB	2D	-	42	B	58	X	6D	m
03	ETX	18	CAN	2E	.	43	C	59	Y	6E	n
04	EOT	19	EM	2F	/	44	D	5A	Z	6F	o
05	ENQ	1A	SUB	30	0	45	E	5B	[	70	p
06	ACK	1B	ESC	31	1	46	F	5C	\	71	q
07	BEL	1C	FS	32	2	47	G	5D	]	72	r
08	BS	1D	GS	33	3	48	H	5E	^	73	s
09	HT	1E	RS	34	4	49	I	5F	-	74	t
0A	LF	1F	US	35	5	4A	J	60	`	75	u
0B	VT	20	space	36	6	AB	K	61	a	76	v
0C	FF	21	!	37	7	4C	L	62	b	77	w
0D	CR	22	“	38	8	4D	M	63	c	78	x
0E	SO	23	£	39	9	4E	N	64	d	79	y
0F	SI	24	$	3A	:	4F	O	65	e	7A	z
10	DLE	25	%	3B	;	50	P	66	f	7B	{
11	DC1(X-ON	26	&	3C	<	51	Q	67	g	7C	|
12	DC2	27	‘	3D	=	52	R	68	h	7D	}
13	DC3(X-OFF)	28	(	3E	>	53	S	69	I	7E	~
14	DC4	29	)	3F	?	54	T	6A	j	7F	DEL
		2A	*			55	U				