900

SERIES
DIGITAL
COMMUNICATIONS

EUROTHERM
CONTROLS

Digital communications handbook

EUROTHERM

900 SERIES

900 Series Digital Communications Handbook

User Guide

HA023776 Issue 2.1

VERSION HISTORY

Version	Date	Changes
1	January 28, 1992	Version 1

Contents

1 Scope

This document is written for people who need to supervise and configure Eurotherm's 900 Series range of communicating controllers, the 902, 903, 904 and 900 EPC (also known as the 905), using a communication link and Eurotherm International's Bisynch standard.

This document is a simple guide illustrating how to use the communication facilities of the 900 Series to configure and supervise 900 Series instruments. It is assumed that the reader has some experience with serial communication protocols. It is also assumed that the reader is familiar with these instruments.

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

Before reading further you might like to consider whether you really want to develop your own supervisory system, when Eurotherm can offer you a range of inexpensive packages, the IPS range, that provide all of the most commonly required features of a supervisory system. Details of IPS are given below.

IPS Product Concept

Instrument Programming System or **IPS** is the name of a new family of software system products which are specifically designed to provide remote access to the Eurotherm 900 series of instruments using an IBM or compatible personal computer (PC) connected to the instrumentation via a serial communications link. Whether on the shop floor, in the plant control room, or in the office, the system will provide many of the features normally only available on expensive supervisory systems. Through careful design of the user interface, using menus and help screens, the **IPS** system is easy to learn and install, even for people with minimum or no computer experience.

IPS is available in three different levels of functionality, tailored to match a wide range of industrial requirements.

IPSL

This is the entry level system[1] which is suitable for small systems of 1 to 4 instruments but provides a significant range of features including:

- Initial assistance with setting up the communications link, for example, identification of instrument addresses is completely automatic.
- Live display of all engineering parameters from any of the four instruments, such as Process value, Setpoint, Integral Time, Autotune status etc.
- Access to update selected engineering parameters, for example, a control loop setpoint can be changed directly from the screen.
- Parameters can be viewed by access level, so that less used parameters can be hidden from view.
- Summary screens are provided for the key parameters from the different instrument control loops such as Setpoint, Process Value, Output.
- New instruments can be cloned from the configuration of another instrument by copying the configuration data over the communications link.
- Information is provided by on-line help screens, removing the need for multi-page user manuals.

[1] Available April 1992

- Instruments that support ramp-dwell programs can be configured from a screen using a simple editor that shows ramp and dwell profiles graphically.
- Segments of ramp-dwell programs can be inserted and deleted without needing to access the instrument front panel.
- Program summary screens provide a concise view of the status of programs in all the instruments, for example, which programs are running, in hold etc.
- To assist with commissioning and process monitoring, the system also includes a historic trending screen so that changes to the process value and setpoint of a control loop can be shown graphically.

IPSG

This is a mid-range system[2] that is suitable for users who need a simple recipe system or need customized views of key parameters from a group of 1 to 4 instruments. This system has all the features of IPSL plus:

- Storage of engineering and configuration parameters extracted from any instrument, in PC files.
- Provision to take the stored parameters held in named files and load to any selected instrument. This allows any configuration or set of engineering working parameters to be quickly changed, for example, when there is a process change.
- Ramp-dwell programs can be saved to named files and then re-loaded any time in the future, to any instrument of the same configuration.
- A user screen can be customized to display any set of parameters from any of the four instruments. A customized screen can be saved and loaded from file so that it can be quickly loaded when the system powers up.
- The parameters displayed on a user screen can be used to create a recipe which can be saved in a file. It is then possible to select a recipe file and load the parameters into the instruments, so providing a very flexible and powerful recipe system.
- Data from a user screen can be logged to a file so that it can be later processed using a package such as Lotus 1.2.3 [3]

IPSX

This top-range system[4] is suitable for users who have a larger number of instruments and have a requirement for multiple user screens associated with complex recipes. Unlike the other systems, **IPSX** supports up to 16 instruments and has all the functionality of the IPSL and IPSG, plus:

- Ability to create multiple customized user screens, so that a set of screens can be tailored for different types of user e.g. operator, process engineer, production supervisor.
- All screens can be saved to a file and rapidly re-loaded on request or automatically when the system powers up.
- Screen access can be protected using a security password.
- Recipes can be defined by being associated with any number of different user screens.
- Instruments can be completely re-configured remotely from special screens which will provide full textual descriptions for all the princpal configuration options. The system will help the user avoid selecting illogical or conflicting configuration options.

[2] Available June 1992
[3] Trade Mark of Lotus Development Corp.
[4] Available August 1992

Ordering

All three systems will be available on both 3.5 and 5.25 inch floppy disks and will run on PCs which have the following specification or better:

PC compatible AT with 1 serial communications port, 640K memory

DOS 3.1 upwards, with CGA, EGA, VGA or Hercules display

Although a mouse pointer is recommended for use with all IPS systems, it is quite easy to use the normal cursor (arrow keys) if a mouse is not available.

In order to preserve investments in IPS, upgrades can be ordered to convert any IPS product to any variant higher in the product family, for example, to upgrade IPSL to IPSX.

For the user who has a high volume requirement for IPS, there is a special discount for ordering systems in sets of 10.

There is also a special offer of a free IPSL system with an order for 10 or more 900 series instruments.

For more information contact your local Eurotherm Sales Engineer or Sales Office.

2 The Bi-synch standard

Eurotherm 900 series communicating controllers are designed to operate with either RS422 or RS422 (485)[5] digital communications; the protocol used corresponds to ANSI X3.28 rev. '76, subcategories 2.5 and A4. Messages consist of ASCII characters including non-printable "control" characters. Briefly the following facilities are available:

- Reading of single parameters by specifying an instrument address and a parameter mnemonic.
- Writing of single parameters by specifying an instrument address and a parameter mnemonic.
- Fast polling of a fixed list of parameters, using a single character message to get the next in a sequence.†
- Reading of many values in one composite message; a fixed set of predefined composites are supported.‡
- Splitting of long messages into a sequence of smaller messages — known as block transmission.‡
- Reading of status words which contain multiple bits of information.
- Broadcast mode whereby an identical single parameter change in a number of controllers in a single network can be achieved.

†Not available in 900 EPC.

‡Only available in 900 EPC.

[5] Although the 900 controllers meet all the drive specifications of the RS485 standard, the Eurotherm system does not support the half duplex clauses within the standard

2.1 Multi Drop Supervisory Link

Transmission Standard RS422 (RS485) (bi-directional)

Protocol ANSI-X3.28-2.5 A4

Data Rates 300,600,1200,2400,3600,4800, or 9600 baud. The 900 EPC will also operate at 19200 Baud, but not at 600 or 3600 Baud.

Character Format 1 start bit, 7 ASCII data bits, 1 stop bit.

Parity Even.

Communications Latency For the 902, 903 and 904 the delay in receiving a response is between 10ms and 125ms. For the 900 EPC single parameter read or writes have a delay of less than 3ms. Multi-block read/writes, or read/writes to status words have a latency of between 50ms and 200ms. For writes to the instrument an acknowledgement will be received at this time if the message was received by the instrument correctly. The message may not have been actioned since it is done at a lower priority than control. To see if the message has been actioned the user must read the error status parameter EE (see section 2.10).

2.2 Single Serial Link

Transmission Standard RS-232 (bi-directional)

Protocol ANSI-X3.28-2.5 A4

Data Rates 300,600,1200,2400,3600,4800, or 9600 baud. The 900 EPC will also operate at 19200 Baud, but not at 600 or 3600 Baud.

Character Format 1 start bit, 7 bit ASCII data bit, 1 stop bit.

Parity Even.

Communications Latency For the 902, 903 and 904 the delay in receiving a response is between 10ms and 125ms. For the 900 EPC single parameter read or writes have a delay of less than 3ms. Multi-block read/writes, or read/writes to status words have a latency of between 50ms and 200ms. For writes to the instrument an acknowledgement will be received if the message was received by the instrument correctly. The message may not have been actioned since it is done at a lower priority than control. To see if the message has been actioned the user must read the error status parameter EE (see section 2.10).

2.3 Physical and Electrical Limits

	RS232	RS422	RS485
Electrical connections	3-wire, single ended	4-wire, differential	
No of drivers and receivers allowed per line	1 driver, 1 receiver	1 driver, 10 receivers	32 drivers, 32 receivers
Maximum cable length	50ft, 15 metres	4000ft, 1200 metres	

2.4 Explanation of terms

2.4.1 Address

Each 900 instrument has a configurable address consisting of two digits, the first being a 'group' number 0 to 9 , the second being a 'unit' number 0 to 9. There are therefore 100 different addresses 00 to 99. This can be configured as described in the handbook for the instrument.

2.4.2 Mnemonics

Mnemonics are two characters that specify which parameter of a control loop is being addressed. For example PV is the process variable and XP the proportional band. Full tables for each instrument are in sections 3, 4 and 6.

2.4.3 Channels

Only the 900 EPC uses channels; in multi-loop instruments, channels are used to identify which loop is being addressed. The 902, 903 and 904 do not use channels as they are single loop. The 900 EPC has two loops on channels 1 and 2 and also a programmer on channel 3.

In making up addresses the channel number is encoded as the ascii character for the channel and precedes the mnemonic. For the 900EPC if the channel number is omitted it defaults to 1. For example the process value on loop two for the 900EPC must be addressed as 2PV, whereas PV on loop one can be addressed as PV or 1PV.

When replying to certain types of messages instruments will sometimes send back the mnemonic and channel if it was specified. The 900 EPC will always transmit the channel number for any mnemonic, even if it received a message with the channel number omitted.

2.4.4 Control characters

Twelve ASCII control characters may be used to delimit messages. These are:

Hex value	Name	Usage
02	STX	Start of data in a message
03	ETX	End of message
04	EOT	End of tranmission sequence
05	ENQ	Enquiry for a value
06	ACK	Positive Acknowledge
15	NAK	Negative Acknowledge
01	SOH	Start of a block transmission
17	ETB	End of block transmission
1C	FS	File separator in composite messages
1D	GS	Group separator in composite messages
1E	RS	Record separator in composite messages
1F	US	Unit separator in composite messages

2.4.5 Data formats

Data in bisync messages is sent as a sequence of ASCII printable characters.

There are five possible data formats associated with each mnemonic (parameter), Free, Fixed, Hex, IEEE or Text.

There is no distinction made in the numeric formats (Free, Fixed and IEEE) between a floating point and an integer value. Integers should be transmitted as a floating point number with no digits after the decimal point, (any digits transmitted after the decimal point are ignored).

Free format All of the 900 series can be set to transmit data in this format by setting a bit in the status word (SW) parameter.

In this system up to six character positions are used. Data may be padded with ascii spaces. A negative number is denoted by a '-' character in front of the number. For example a value of 13.9 could be sent as any of the following, where □ denotes the ascii space character (hex 20),

0013.9
□□13.9
13.9
□13.9□
013.90

and −2 could be sent as any of

−2.0
□□ − 2.0
□ − 2.0□
−2.000
−02.000
−2□□
−2

Note that the 900 series will always use the most compact form of the data when replying to reads; it will accept data in any valid free format.

Fixed format This format is not supported by the 900 EPC. The 902, 903, and 904 can be set to transmit all numeric data in this format by setting a bit in a status word (SW) parameter.

It uses five character positions which must all be filled. For negative values the decimal point is replaced by a minus sign. So for example 5.3 can be entered as

5.300
05.30
005.3

and −5.3 as

5 − 300
05 − 30
005 − 3

Hex format Here the data is preceded by a '>' (hex 3E), and normally consists of four hex characters, (negative values cannot be transmitted). This format is typically used to write bit patterns into status words or to read status words. It is also used to read or write the block length (BL) mnemonic to the 900 EPC, and to read software version ids and instrument ids from all the 900 series instruments.

Each data character is transmitted as an ascii hex value (i.e. 0 to 9 and A to F). The hex value is converted to a bit pattern, and sets the bits in the status word accordingly, so that 0 clears 4 bits, 1 sets the rightmost bit, and so on, up to F which sets all four bits, as shown in the table below.

Hex value	Bit pattern
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

The first data byte in the message sets the leftmost bits in the status word. So a message containing the four characters F000 would set a 16 bit status word to 1111000000000000.

One exception to the four hex character rule is the WI and WO mnemonics for the 900 EPC; these mnemonics use eight hex characters since they are 32 bit status words. The format of the data is the same, the first hex character sent sets the leftmost 4 bits in the 32 bit status word.

When used in the 900 EPC to read and write BL the resulting bit pattern is treated as two binary numbers (unsigned) which give the values of the block length for the 900 EPC and the supervisory machine. The first two hex characters used, give the block length for the supervisor machine talking to the 900 EPC, the last two are read only and give the value of the block length on the 900 EPC; this is always FF (255). The supervisor may set the first two bytes to any value greater than or equal to hex 10 (16) and less than or equal to FF (255). The default for BL is 80FF i.e. the supervisor has block length of 128.

One other use for hex format is on the 900 EPC where it is possible to make parameters that appear on the instrument display parameter scroll list for each loop read only, read or write, or disappear from the list, by writing a hex format message with one data byte e.g `PV>2` if the mnemonic is PC. If the value is 2 the parameter will no longer appear on the scroll list, 1 it will be read only, 0 will make it read and write. Any other value is invalid.

IEEE format This is only supported when writing to 902, 903 and 904 but can be used for reading from or writing to 900 EPC. The 900 EPC can be set to send numeric floating point data in IEEE format using a bit in the status word (SW).

The data is preceded by the `@` (hex 40) character. It consists of up to six bytes of packed single-precision IEEE format floating point data. In most cases, up to 4 bytes are used to represent 24 bits of the IEEE word, the least significant 8 bits are ignored as they define a greater precision than is normally required. However, where higher precision is required, the full 32 bits can be encoded into 6 bytes.

Each of these fields are packed into a byte using bit 5 through bit 0, (the numbering of bits is shown in figure 1). To ensure that bytes only represent printable ASCII, bit 6 is always set to 1. Bit 7 is used for parity as normal.

Field	Field Bits	Bits from IEEE Single Precision word
1	5-0	31 - 26
2	5-0	25 - 20
3	5-0	19 - 14
4	5-0	13 - 8
5	5-0	7 - 2
6	5-4	1 - 0

BIT	31	30 - - 23	22 - - - - - - - - 0
	S	2^7 … 2^0	2^{-1} 2^{-2} … 2^{-23}
		exponent	fraction

IEEE format for floating point storage in 32 bit words.

for $0 < E < 255$ value $= (-1)^s * 1.F * 2^{E-127}$ (Normalised form)

$E = 0$ and F not equal 0 value $= (-1)^s * 1.F * 2^{-126}$ (Denormalised)

BIT 7 *parity*							BIT 0	ORDER OF TRANSMISSION
P	1	S	2^7	2^6	2^5	2^4	2^3	1
P	1	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2
P	1	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	3
P	1	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	4

Figure 1: IEEE Floating Point Number Format

If field 6 is transmitted, the lower nibble is always zero and is ignored.

The packing of bits into fields and order of transmission is shown in figure 1.

If the last field to be transmitted has all the encoded bits at 0, then the field need not be transmitted. For example if all the bits 6 through 0 of field 4 are zero, then field 4 need not be transmitted.

The same rule applies to each last field in turn, so that 4,3,2 or 1 fields can be transmitted if the trailing bits are zero. The receiver should re-constitute the IEEE number by assuming that non-transmitted fields contain all zero bits.

In the most degenerate form, an IEEE encoded value of 0 can be transmitted by suppressing all fields and only sending the "@" prefix.

The suppression of zero fields to improve comms. bandwidth is optional; it is acceptable to send all 4 or 6 bytes. The instrument will accept the suppressed or full encoded form of up to 6 bytes.

Text format This is used read and write strings for the display on the 900 EPC. The data is preceded by a ' (hex 27) character and then consists of any number of printable ascii characters.

2.5 Reading data from the 900

Figure 2 shows the sequence used for reading data from an instrument.

Enquiry To read data a read message is issued to the instrument. For the 902, 903 and 904 series the message consists of

```
[EOT] (GID) (GID) (UID) (UID) (C1) (C2) [ENQ]
```

where [XXX] denotes the control character XXX, (i.e. [EOT] is the EOT control character), (GID) denotes the group number, and (UID) the unit number (see section 2.4.1). This sequence selects the instrument from all others on the network.

The (C1) (C2) is the two character parameter mnemonic (see section 2.4.2) and selects the parameter to be read.

The 900 EPC is a "multi-channel" instrument because it has two loops and a programmer, (see section 2.4.3). Some of the mnemonics require the loop being read from to be identified using an extra channel number digit, placed before the mnemonic in the message, viz:

```
[EOT] (GID) (GID) (UID) (UID) (CN) (C1) (C2) [ENQ]
```

(CN) selects the loop, if it is ascii 1, loop 1 is selected; if it is 2, loop 2 is selected, 3 selects the programmer. If it is omitted, loop 1 is selected.

All 900 EPC loop dependent parameters have a channel number, and all 900 EPC programmer related mnemonics must be addressed using channel number 3.

Step 1 in figure 2 is the enquiry step, where <ADDR> is the (GID) (GID) (UID) (UID) sequence, and <MNE> is the (C1) (C2) or (CN) (C1) (C2) sequence.

Valid response If the instrument receives the message correctly and the mnemonic (and channel number if provided) is valid then it will reply with

```
[STX](C1)(C2)<DATA BYTES>[ETX]<BCC>,
```

for the 902, 903 and 904 or

```
[STX](CN)(C1)(C2)<DATA BYTES>[ETX]<BCC>,
```

for the 900 EPC where (CN) is the selected loop number (if the loop number was omitted in the original request it is still sent back as the default loop 1).

The data bytes are as described in section 2.4.5. <BCC> is a block checksum that is used to validate the data. It is computed by XORing (exclusive or) all the characters after and excluding [STX] and up to and including [ETX]. (Step 2 in figure 2).

Invalid response If the reply is invalid (either the block sum check is wrong, the received mnemonic is not the one asked for, or the data bytes are in an invalid format) then the whole read request can be retransmitted (step 7 in figure 2) or for the 902, 903 and 904 instruments only a [NAK] sent, which will cause the instrument to retransmit the reply, (step 4 in figure 2).

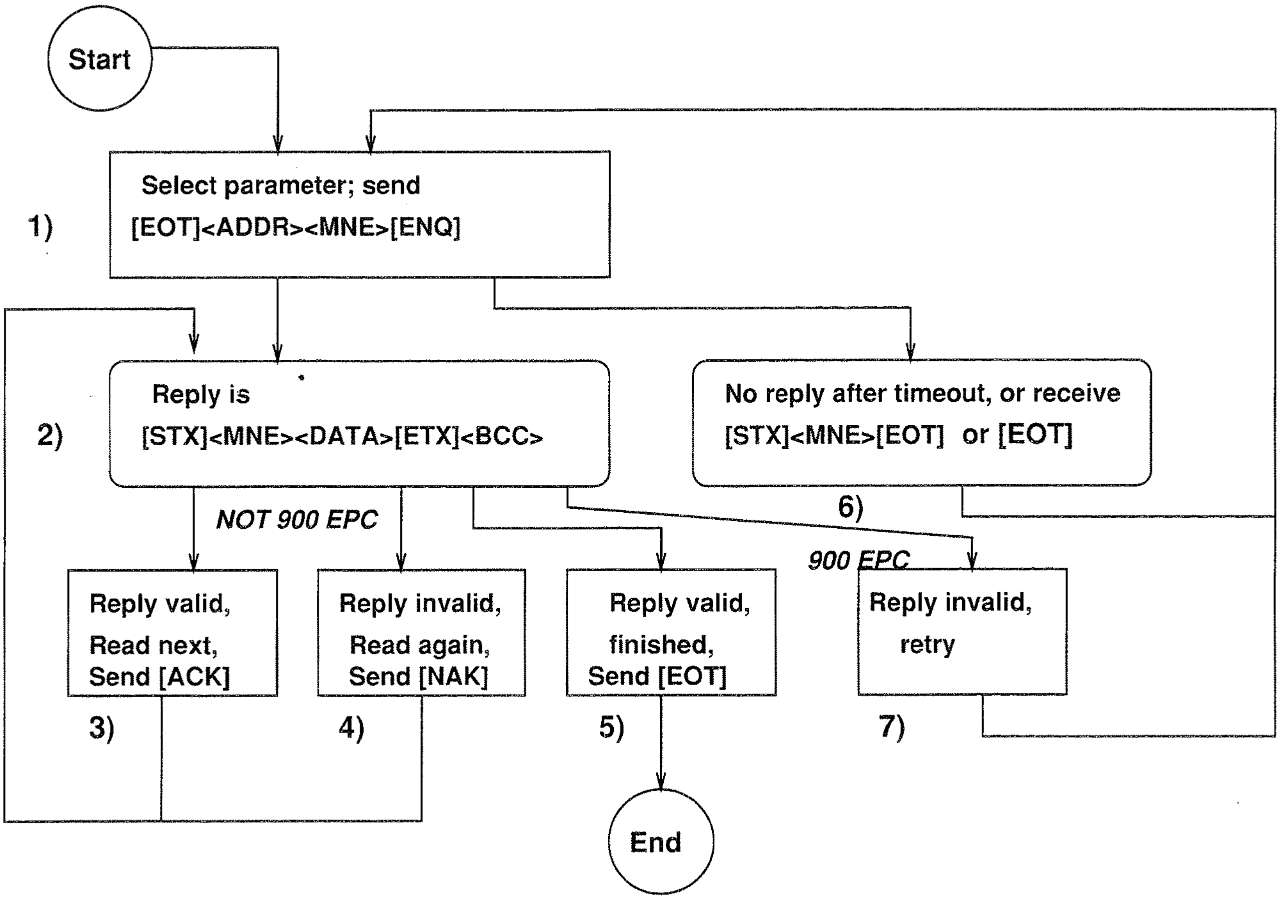

Figure 2: Reading Data from the 900 series

No reply or request rejection If no reply or only a partial reply is received after a suitable time out period then the read request should be retried (step 6 in figure 2). The instrument may not reply if

- The baud rate is set wrongly
- There was a parity error or other error in a character received
- Noise may have corrupted the message
- There may be a hardware fault
- The group and/or unit address identifiers may not have been recognised

The timeout period should be computed from the baud rate and the latency of the instrument. The timeout should be reset once a character has been received. The reader should wait for the timeout period before retrying a read request. (For latencies see sections 2.1 and 2.2).

If the mnemonic received by the instrument is not valid, the instrument will reply with

`[STX] (C1) (C2) [EOT]`

or simply

`[EOT]`

for the 900 EPC.

The read request can then be retried (step 6 in figure 2) or another mnemonic chosen.

Fast poll list If the reply is valid, the 902, 903 and 904 series support a read next parameter "fast poll" or "scroll mode" facility. An instrument, once selected, will return the values of a list of parameters; the next parameter in the list can be read by sending an `[ACK]` (step 3 in figure 2). This list can be treated as a "circular buffer" of parameters so that eventually the first parameter read will be re-read.

Termination If the reply is valid and no more action is to be taken, an `[EOT]` should be sent to deselect the instrument (step 5 in figure 2). Sending of the `[EOT]` can be omitted if the next action is to select another instrument because an `[EOT]` will be sent then (step 1 of figure 2).

2.6 Writing Data to the 900

The sequence for writing data to the 900 is shown in figure 3.

Connect and write data The write message consists of two parts, first the establish connection, (known as selecting the instrument).

[EOT] (GID) (GID) (UID) (UID)

where (GID) denotes the group number, and (UID) the unit number (see section 2.4.1). This sequence selects the instrument from all others on the network. (Step 1 in figure 3, <ADDR> is the (GID) (GID) (UID) (UID) sequence).

This should be immediately followed by the mnemonic identifier and the data to be written

[STX](C1)(C2)<DATA BYTES>[ETX]<BCC>,

where the data bytes are as described in section 2.4.5. <BCC> is a block sum check that can be used to validate the data. It is computed by XORing (exclusive or) all the characters after and excluding [STX] and up to and including [ETX]. (Step 2 in figure 3, where <MNE> is the (C1) (C2) sequence).

For the 900 EPC a channel number may also be specified for loop specific parameters, to select channel 1 or 2, or channel 3 for 900 EPC programmer related parameters, viz

[STX](CN)(C1)(C2)<DATA BYTES>[ETX]<BCC>,

where (CN) may be ascii 1 for loop 1 or 2 for loop 2 or 3 for the 900 EPC programmer. If it is omitted it defaults to loop 1.

Note that the part of the message containing the data is identical to the reply from the 900 to a read request (section 2.5), (except if the message has been written to the 902, 903 or 904 in IEEE format which only send data in fixed or free format).

Responses from the 900 When the message is received it will be ignored if it contains parity errors, otherwise the 900 will

- Verify the BCC is correct, if it is then ...
- Verify that (C1) (C2) or (CN) (C1) (C2) command characters are a valid mnemonic that may be written to, if so ...
- Verify that the data is in a valid format for the parameter and is not out of range, if so ...
- send an [ACK] as a reply (step 2 in figure 3).

If any of the above steps fail then a [NAK] will be sent by the 900,(step 4 in figure 3).

On receiving an [ACK] the data has been successfully written. To send more data to another parameter (or the same one) another data message may be sent (step 2 in figure 3).

To terminate the connection an [EOT] should be sent to the 900 (step 5 in figure 3).

On receiving a [NAK] the write may be retried after a timeout period (but if the error is due to out of range, or invalid mnemonic it will fail again).

No reply may be received if

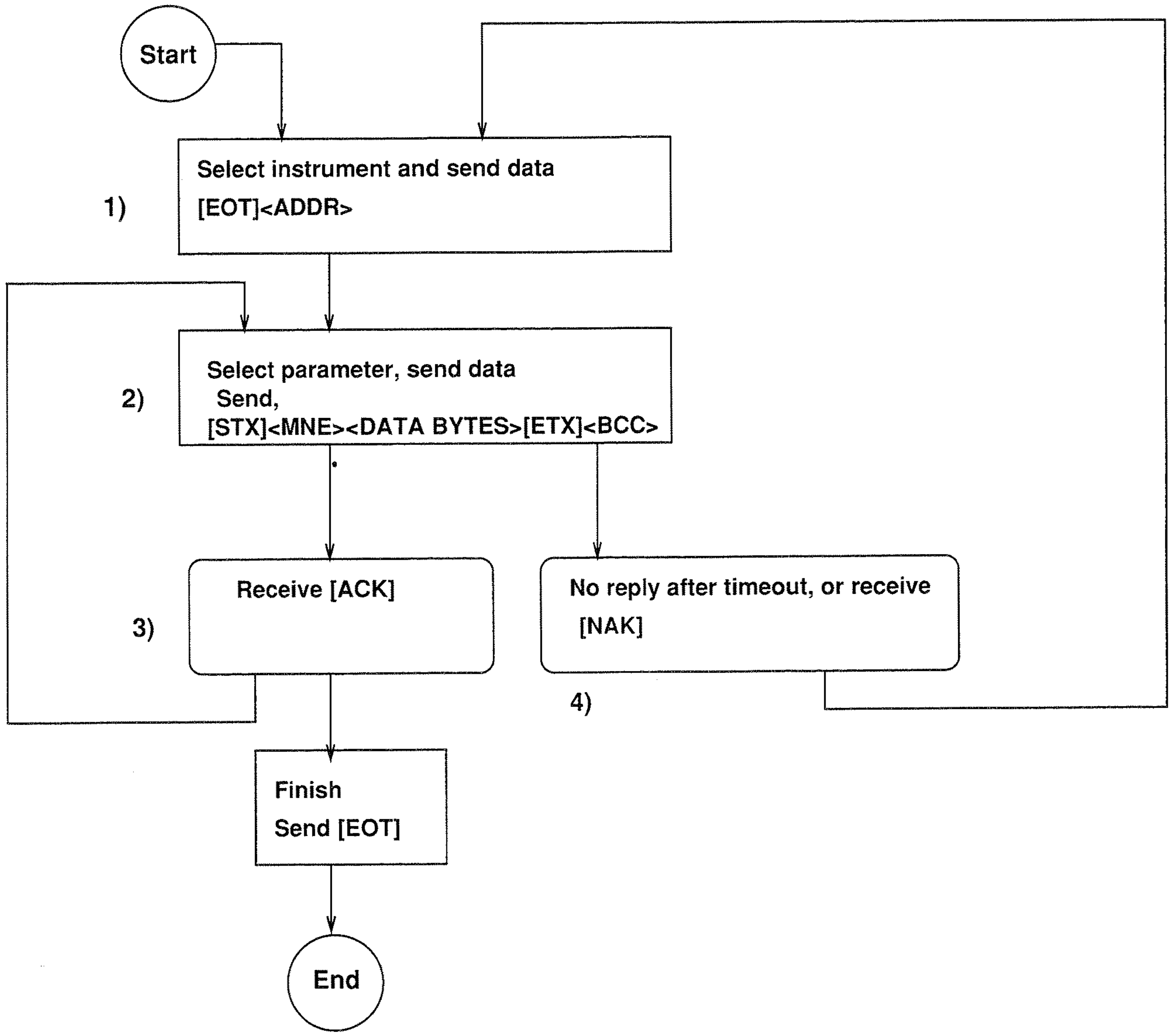

Figure 3: Writing data to the 900 series

- The baud rate is set wrongly
- There was a parity error or other error in a character received
- Noise may have corrupted the message
- There may be a hardware fault
- The group and/or unit address identifiers may not have been recognised

The timeout period should be computed from the baud rate and the latency of the instrument. The timeout should be reset once a character has been received. The writer should wait for the timeout period before retrying a write request.

2.7 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.

or to specific groups of instruments.

A broadcast takes the form of normal write with a selection of the instruments using the GID and UID, and then data transfer (see section 2.6). However, unlike a normal "write", the instruments receiving the broadcast message will not respond (neither with an [ACK] nor [NAK] character). If a receiving instrument detects an error in a broadcast message, the mnemonic EE will be set accordingly but a [NAK] will not be sent, so to find if the broadcast suceeded the EE mnemonic will have to be read individually for each instrument.

The address GID = ~, UID = ~ can be used to broadcast to all multi-dropped instruments.

However, when a GID is specified and the UID = ~ , then only instruments in the given group will receive the broadcast.

For example, GID = '1', UID = ~ can be used to broadcast to all instruments in the group 1.

If the GID = ~ and UID is specified, then the broadcast is received by all instruments with the same UID.

For example, GID = ~, UID = '2', then all instruments with UID = 2 will receive the broadcast.

An instrument may provide a mechanism to switch the reception of broadcast messages ON or OFF for safety or other reasons.

No more communication transactions should be allowed after a broadcast message, until sufficient time has elapsed for all instruments to have dealt with the broadcast message.

2.8 Block transmissions

The 900 EPC supports very long composite messages (see section 2.9); it is possible that a message may exceed the maximum length supported by an instrument or indeed the computer talking to an instrument. In this case a "transport layer" may be used which splits up one large message into several smaller sub-messages for transmission and allows the full message to be re-assembled.

The maximum message length is controlled by the value of the mnemonic BL; if a message exceeds this length it must be split into sub-messages, (see section 2.4.5 under hex format messages for how BL is set).

Sub-messages are essentially the same as messages containing data, but they have an added header character [SOH], followed by a block number, and then instead of [ETX] an [ETB] character, as shown below,

```
[SOH]<BLK>[STX]<TRANSMISSION BLOCK>[ETB]<BCC>
```

where the number of characters in the transmission block i.e. from the first character after [STX] to the last character before [ETB] is the stored in the parameter BL.

The character <BLK> is a block number for each sub-message, which starts at 1 for each new message and then is incremented modulo 8 for each subsequent block (i.e. the sequence is 1 2 3 4 5 6 7 0 1 2 3 4 5 6 . . .).

<BCC> is the bitwise exclusive or of all the characters from after the [SOH] up to and including the [ETB].

Only the first sub-message received from an instrument will repeat the mnemonic and channel identifier, so it will have the form

```
[SOH][1][STX](CN)(C1)(C2)<DATA BYTES>[ETB]<BCC>
```

The last message in a series is denoted by the [ETX] character instead of the [ETB] at the end of the message, i.e.

```
[SOH]<BLK>[STX]<DATA BYTES>[ETX]<BCC>
```

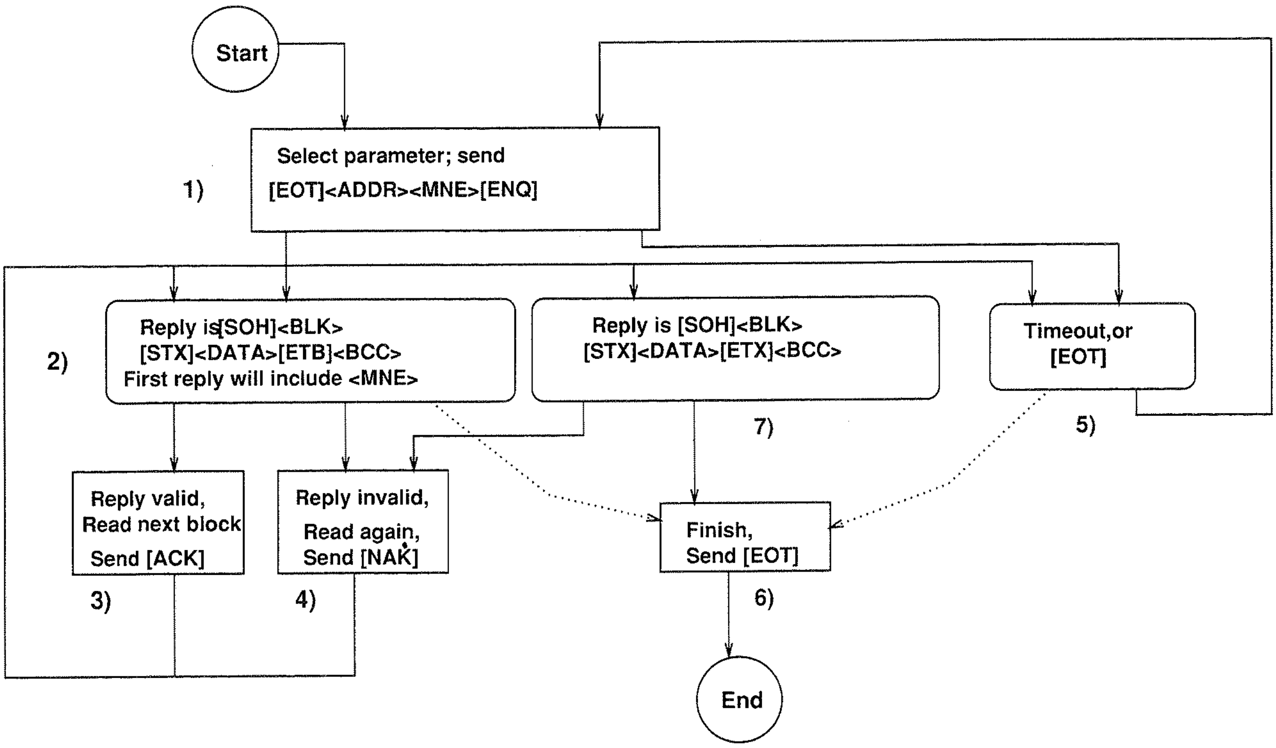

Figure 4: Reading long messages from the 900 EPC

In step 1 in figure 4 the reader sends the instrument the usual sequence for selecting an instrument, a channel if required, and a mnemonic (see 2.5).

In step 2 in figure 4 the reader receives a reply that begins with [SOH]; this tells the reader that the message will be divided into blocks. The next character is the ascii block number which should start at one, and increase modulo 8 on subsequent messages.

If the message is valid (i.e. the whole of the message is received before a timeout and the BCC block sum check is valid and the block number is the next in sequence) the reader requests the next message by sending an [ACK], step 3 in figure 4. This process is repeated until the last message is received; the reader knows it is the last message because it is terminated by an [ETX] character, (step 7 in figure 4). The reader then sends an [EOT] (step 6 in figure 4) to terminate the connection to the instrument.

If at any stage a sub-message is received and is not valid (the block sum check fails or the block number is wrong) the reader can request retransmission using a [NAK], (step 4 in figure 4). If at any stage the reader receives [EOT] or a request times out then the reader can start again, or disconnect (step 5 in figure 4). If at any stage the reader decides to terminate, an [EOT] can be sent to abort the sequence of messages and disconnect from the instrument (shown by the dotted line transitions to step 6 in figure 4).

The sequence for writing a long message to the 900 EPC is the reverse of the above. It is shown in figure 5.

Step 1 is to select the instrument by sending the sequence

```
[EOT] (GID) (GID) (UID) (UID)
```

Then on step 2 the first data block with block number 1 is sent. This data block may also contain the channel number (if omitted this defaults to channel 1) and must contain the mnemonic. So it will look like

```
[SOH][1][STX](CN)(C1)(C2)<DATA BYTES>[ETB]<BCC>.
```

If an [ACK] is received (step 3 in figure 5) then the next block can be written. Again the block number is incremented modulo 8. Subsequent blocks do not contain the mnemonic and channel number.

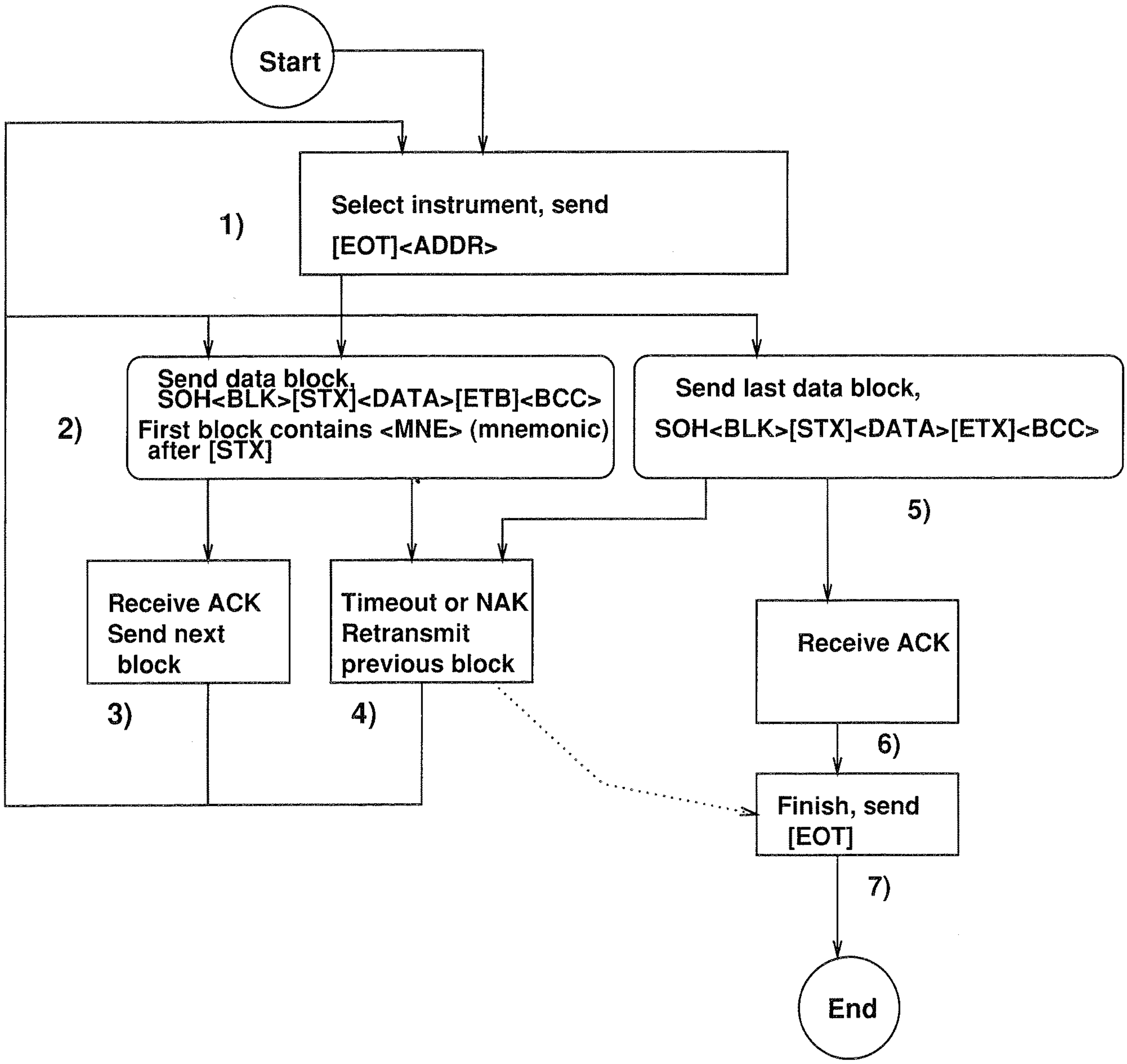

Figure 5: Writing long messages to the 900 EPC

This process is repeated until the last block is sent (step 5 in figure 5). If an `[ACK]` is received (step 6 in figure 5) then the writer sends `[EOT]` to terminate the message (step 7 in figure 5).

If a `[NAK]` is received at any point or no `[ACK]` is received after a timeout (step 4 in figure 5) then the writer can resend the previous transmission; if it is the first transmission then the whole of steps 1 and 2 must be repeated. Alternatively an `[EOT]` can be sent to the instrument to abort the write.

The final `[ACK]` after the last message block is sent means that the instrument has successfully received the message with no BCC errors; it does not mean that the instrument has implemented the write (this differs from normal writes where the `[ACK]` means the message was received, understood and implemented). When writing using block messages, to find out whether the write was successful the writer must read the value of EE (see section 2.10).

2.9 Composite data

Composite data is a way of sending many values in one message. Usually the message is so long that it must be divided into sub-messages using the technique described in section 2.8.

Composite data in a message is divided into records. Each record can itself be divided into sub-records; this hierarchy of records can be continued to 3 levels.

The characters FS (hex 1C), GS (hex 1D), RS, (hex 1E) and US (hex 1F), (see section 2.4.4) are used to delimit records and sub-records. FS is the highest level separator, then GS, then RS and finally US.

The first character in the composite data (which will be embedded in a bisync message like any other data) will be one of FS, GS, or RS and indicates how many levels there are to the records in the message. It will be the highest level separator in the message; subsequent characters are used to separate each level as shown in the table below:

Abbreviation	Hex Code	Number of levels in message, when first char	Level separated when not first char
FS	1C	3 levels	
GS	1D	2 levels	Highest level separator
RS	1E	1 level	Middle (if FS appears) or highest (if FS does not appear) level separator
US	1F		Lowest level separator

The occurence of a higher level separator implies the simultaneous occurence of all lower level separators.

For example, in the following description . . . represents an encoded data field of any EI Bisync data format as described in section 2.4.5 with the exception of Composite Data.

A single level stream of data fields will be encoded as

`RS . . . US . . . US . . .`

RS introduces a single stream of fields separated by US; the first US is implied.

Composite data that consists of a series of single-level records each of which contains three fields is encoded as

`GS . . . US . . . US . . . RS . . . US . . . US . . . RS . . . US . . . US . . .`

GS introduces a series of single-level records, the first RS and US is implied.

Composite data that consists of a series of two-level records, each record containing two sub-records, each of which contains two fields, is encoded as

`FS . . . US . . . RS . . . US . . . GS . . . US . . . RS . . . US . . .`

FS introduces a series of two-level records, the first GS, RS and US is implied.

As mentioned above composited data is embedded in a bisync message. A particular composite data item is associated with a mnemonic and channel. To read and write it a standard enquiry may be sent as described in sections 2.5, 2.6 and 2.8. The composite data is extracted from the bisync message in the same way as any other data, and then validated. The formats of all the 900 EPC composites are described in section 6.

All composite data mnemonics have a fixed structure. Sometimes a particular field or whole records may be missing if the option is not configured in the instrument. In this case the field is left blank but the field and record separators are still transmitted.

All composites and status words in the 900 EPC are read and written in the "background"; in other words they have a long communications latency and control takes priority over communications. When a write to a composite is acknowledged it means that the message was received correctly. It does not mean that the message has been implemented. The message is not implemented until the value of EE is zero. So writers should write the composite, and then read EE until it is zero, before doing any other write of composites or status words, (see section 2.10).

2.10 Error codes held in EE

The EE parameter can be read to give the status of the last communications transaction. The values common to all the 900 series are shown in the table below.

900 Series EE values		
Error	Hex Value	Meaning
No error	0000	
Invalid Mnemonic	0001	
Read Only Error	0004	The parameter cannot be written
Invalid Data Format	0007	For 902,903 and 904 this includes BCC errors
Limit Error	0008	The value written is out of range

The 900 EPC defines some additional values for EE, shown in the table below.

Additional EE values in the 900 EPC		
Error	Hex Value	Meaning
BCC Error	0002	
Invalid Channel	0006	
End of buffer	0014	The message islonger than the BL parameter allows
Invalid block number	0015	A block number in a sub-message is out of sequence
Background Active	0017	A command that has a long latency is still being processed [2]
Bad Uid	001F	The second UID did not match
Bad Gid	0020	The second GID did not match
Calibration Not Ready	0021	Calibrate command being processed
Parameter Not Configured	0022	
Comms Disabled	0032	All writes are disabled
Invalid Status Message	0024	
Unknown Status Mnemonic	0025	
Parameter not scrolled	0027 [1]	
Invalid Separator for Composite Data	0028	
Background Write In Progress	0046	A write message divided into sub-blocks was received o.k and is now being actioned [2]

There are also additional values that occur when there are errors in accessing the programmer over communications using channel 3.

[1] It is possible to make parameters that appear on the instrument display parameter scroll list read only, read or write, or disappear from the list, by writing a message of the format `[Mnemonic]>[Value Byte]` (i.e. a hex format message with one data byte). If the value is 2 the parameter will no longer appear on the scroll list, 1 it will be read only, 0 will make it read and write. If this command is sent for a parameter not on the scroll list, then EE takes the value 39.

[2] For the 900 EPC reading or writing to status words or composites has a long latency. In particular writes are acknowledged by an ACK before they are implemented, i.e. when the message has been successfully received. The writer should not perform another write to the parameter until EE is clear.

Programmer related EE values in the 900 EPC		
Program Segment Not Found	41	
Unknown Program Mnemonic	42	
Invalid Program Number	43	
Not Enough Free Segments	44	
Invalid Segment Type	45	
Invalid End Type	46	
Program Busy	47	
Function Not Confirmed	48	The instrument is not configured for the requested action

2.11 Differences between 902, 903, 904 and 900 EPC

	902,903,904	900 EPC
Channel Number	None	1,2,3
Block Mode	No	Yes
Fast Poll	Yes	No
Composite Data	No	Yes
Fixed Format Data	Yes	No
Free Format Data	Yes	Yes
Ieee Data	Write only	Yes
Text String Data	No	Yes
Comms Latency	10-125ms	<3ms single parameter 50-200ms multi-block and status word
Baud Rate	300-9600 Baud	300-19200 Baud excluding 300 and 3600
Broadcasts	Yes	Yes

3 902, 903, and 904 mnemonics

3.1 Instrument Identity — II

The II mnemonic gives the instrument identity, 9020 for all 902, 903 and 904 instruments.

3.2 Software Version — V0

This has four data bytes, the first giving the software class, the second the software version and the last two the issue of the software.

3.3 Instrument Mode — IM

The 902-4 will allow configuration to be entered via digital communications. In configuration mode, outputs are disabled and PV measurement is suspended.

```
Instrument Modes:

IM    = 0   Normal Operation Mode
      = 1   No effect
      = 2   Configuration Mode
```

Warning: Once in configuration mode, other test functions are available via IM. Writing a value other than 0, 1 or 2 may cause all calibration and configuration information may be lost.

3.4 Comms Status — EE

This returns a value 0 when the previous comms transaction was valid. On errors the error codes are

0	Clear
1	Invalid mnemonic
4	Read only error
7	Data format error
8	Limits error

3.5 902, 903 and 904 Parameter Mnemonic List

This section shows the mnemonic list, split up into the following three tables.

The order of the parameters in this list is that that would be obtained if a fast poll were done starting from parameter II. This is a complete list of all the mnemonics; the actual list for a specific controller will depend on the instrument configuration and will be a subset of this list. The order that parameters appear here will still be adhered to in a fast poll.

Status words are transmitted using hex format and the '>' character (see section 2.4.5). Other parameters may be transmitted using numeric free or fixed format, or may be written using IEEE format. Exceptions to this are given in the above sections.

(R/O) in the list below means that the parameter is read only.

MNEMONIC	PARAMETER	AVAILABILITY
II	Instrument Identity	Always available [1]
V0	Software Version	Always available [2]
IM	Instrument Mode	Always available [3]
EE	Comms Status	Always available (R/O)[4]
1H	Display maximum	Always available [5]
1L	Display minimum	Always available [5]
PV	Measured Value	Always available (R/O)
SP	Working Setpoint	Always available (R/O)
OP	Output	Always available [6]
SW	Status Word	Always available [7]
OS	Optional Status Word	Always available [7]
XS	Extended Status Word	Always available [7]
1A	Alarm 1	Configuration dependent
2A	Alarm 2	Configuration dependent
ER	Error	Always available (R/O)
SL	Internal setpoint (SP 1)	Configuration dependent
S2	Setpoint 2	Configuration dependent
RT	Local Setpoint	Configuration dependent
MP	VP Pot Value	Configurable (R/O)
RI	Remote Input	Remote input other than Power Limit (R/O)
O1	Status Word 1	Programmer and digout configured
O2	Status Word 2	Programmer and digout configured
O3	Status Word 3	Programmer and digout configured
O4	Status Word 4	Programmer and digout configured
O5	Status Word 5	Programmer and digout configured
O6	Status Word 6	Programmer and digout configured

[1] See section 3.1
[2] See section 3.2
[3] See section 3.3
[4] See section 3.4
[5] The true limits of PV are $1H + .1(1H - 1L)$ to $1L - .1(1H - 1L)$.
[6] In Auto Mode output power is read only; in Manual Mode output power is read/write. For a VP, OP is available as diagnostic read only
[7] See section 3.6

902, 903 and 904 Programmer The 902P has one program, the 903P up to 4, and the 904P up to 15. The 902S has no programmer. The following procedure may be used to determine over a communications link which instrument is connected,

- Read mnemonic CP. If it is not available the instrument is a 902S.
- If it is zero the instrument is a 902P.
- Try to write 15 to it. If this fails the instrument is a 903P, otherwise it is a 904P.

With the instrument in operator mode it may be programmed over the communications link. For the 903 and 904 the program must be selected by writing the program number to CP. Once selected the program is available using the ramp rate mnemonics r1 to r8, the ramp level mnemonics l1 to l8 and the dwell time mnemonics t1 to t8. Note that Hb, holdback value, is defined per program.

Mnemonics r1 to r8 and t1 to t8 have certain negative values to indicate a Step (an infinite ramp rate), None (the segment will be omitted), or End (the sequence of segments is terminated). The value read depends upon the main display decimal position as shown below for free format data:

Mnemonic	Action	Decimal Point Position			
		XXXXX	XXXX.X	XXX.XX	XX.XXX
r1-r8	Step	0	0.0	0.00	0.000
r1-r8	None	-1	-0.1	-0.01	-0.001
r1-r8	End	-2	-0.2	-0.02	-0.002
t1-t8	End	-1	-0.1	-0.01	-0.001

The value written may be simply -1, -2 or 0 regardless of the display.

For the 902P Lc is the number of times the single program is sequenced through.

For the 903 and 904 setting mnemonic Lc to zero means go on to the next program. Otherwise if Lc is set to a value the instrument will loop Lc times through the sequence of programs just completed.

In Ramp segments, mnemonics r1 to r8, the mnemonic time remaining, TM, is read only. In Dwell segments, mnemonics t1 to t8, TM is Read/Write.

The following table summarises the programmer parameters (this is still one of the tables in the total mnemonic list that defines the fast poll list).

MNEMONIC	PARAMETER	AVAILABILITY
CP	Current Program Number	Multi Programmer
TM	Time left in current program segment	Programmer active OR Ramp function active (R/O except in hold)
LR	Loops remaining for current program	Programmer active (R/O except in hold)
r1	Ramp rate 1	Programmer
l1	Ramp level 1	Programmer
t1	Dwell time 1	Programmer
r2	Ramp rate 2	Programmer
l2	Ramp level 2	Programmer
t2	Dwell time 2	Programmer
r3	Ramp rate 3	Programmer
l3	Ramp level 3	Programmer
t3	Dwell time 3	Programmer
r4	Ramp rate 4	Programmer
l4	Ramp level 4	Programmer
t4	Dwell time 4	Programmer
r5	Ramp rate 5	Programmer
l5	Ramp level 5	Programmer
t5	Dwell time 5	Programmer
r6	Ramp rate 6	Programmer
l6	Ramp level 6	Programmer
t6	Dwell time 6	Programmer
r7	Ramp rate 7	Programmer
l7	Ramp level 7	Programmer
t7	Dwell time 7	Programmer
r8	Ramp rate 8	Programmer
l8	Ramp level 8	Programmer
t8	Dwell time 8	Programmer
Hb	Holdback value	Programmer with hold back
Lc	Loop count	Programmer
RR	Ramp rate	Ramp function

The rest of the mnemonics are given in the next table.

MNEMONIC	PARAMETER	AVAILABILITY
HO	Max. Heat	Controller with PID Heat
LO	Max. Cool	Controller with PID Cool
RH	Remote Heat Limit	PID Heat + Remote as Heat Limit (R/O)
RC	Remote Cool Limit	PID Cool + Remote as Cool Limit (R/O)
HS	Setpoint 1 Maximum	Always available (R/O)
LS	Setpoint 1 Minimum	Always available (R/O)
H2/TH	Setpoint 2 Maximum	Configuration dependent [9] (R/O)
L2/TL	Setpoint 2 Minimum	Configuration dependent [9] (R/O)
H3	Local Setpoint Max	Remote i/p (R/O)
L3	Local Setpoint Min	Remote i/p (R/O)
2H	Remote Max Scaler	Remote i/p (R/O)
2L	Remote Min Scaler	Remote i/p (R/O)
CH	Cycle Time for Channel 1	Channel 1 time proportioning
XP	Proportional Band	PID heat
TI	Integral Time	PID heat
MR	Manual Reset	On/Off, P or PD controller
TD	Derivative time	PID heat
HB	Cutback High	PID heat
LB	Cutback Low	PID heat
RG	Relative Cool Gain	PID heat/cool
P2	Proportional Band [10]	PID heat /Dual PID
I2	Integral Time [10]	PID heat /Dual PID
R2	Manual Reset [10]	On/Off or P or PD/Dual PID
D2	Derivative Time [10]	PID heat /Dual PID
G2	Relative Cool Gain [10]	PID heat/cool /Dual PID
HC	Heat Cool Deadband	Heat/cool
CC	Cool Cycle Time	Heat/cool and time proportioning O/P 2
C2	Channel 2 Cycle Time	Heat with dual O/P +time proportioning on O/P 2
TT	Travel Time	VP
Tt	Travel Time Down	VP with asymmetric valve
MT	Minimum on Time	VP
TP	Valve Update Time	VP
LE	Low Valve Limit	VP
EH	High Valve Limit	VP
PE	Emissivity	Pyrometer
BP	Power Level at Sensor Break	Always available
TR	Adaptive Tune Trigger	Adaptive Tune

[9] When configured as a remote setpoint H2 and L2 are available. When configured as a remote and local setpoint, TH and TL are available.

[10] These are the second PID parameters; the second PID is selectable using the XS status word (section 3.9).

3.6 Status Word Formats

Status words always use hex format, using four hex characters, denoted below by digits ABCD. These are ASCII characters, representing a hexadecimal digit (0-9, A-F). The binary equivalent of the character in 'A' position gives the values of bits 15 to 12 in the status word, character 'B' bits 11 to 8, character 'C' bits 7 to 4 and character 'D' bits 3 to 0. So for example a message of FFFF would turn all the bits in the status word on.

3.7 Status Word — SW

Digit	Bit	Function	Access	Clear/Set
A	15	Manual Active	R/W	Auto/Manual
A	14	Remote Active	R/W	Local/Remote
A	13	SP2 Active	R/W	SP1/SP2
A	12	AL1 or AL2	R/O	No Alarm/Alarm 1 or 2
B	11	N/A		
B	10	Alarm 1 State	R/O	On/Off
B	9	N/A		
B	8	Alarm 2 State	R/O	On/Off
C	7	N/A		
C	6	N/A		
C	5	Parameter Change	R/O	Not Changed/Changed
C	4	N/A		
D	3	N/A		
D	2	Key Lock	R/W	No/Yes
D	1	T/C Break	R/O	No/Yes
D	0	Data Format	R/W	Free/Fixed

3.8 Optional Status Word — OS

Digit	Bit	Function	Access	Clear/Set
A	15	Digital Input 1	R/O	Off/On
A	14	Digital Input 2	R/O	Off/On
A	13	Dig Output Channel 3	R/W	Off/On
A	12	Dig Output Channel 4	R/W	Off/On
B	11	Segment No. (MSB)	R/O	see [1]
B	10	Segment No.	R/O	see [1]
B	9	Segment No.	R/O	see [1]
B	8	Segment No. (LSB)	R/O	see [1]
C	7	Digital I/P Inhibit	R/W	None/Inhibit
C	6	Ramp/Dwell	R/O	Ramp/Dwell
C	5	Skip Segment	R/W	Remain/Skip
C	4	Hold Logged	R/W	Continue/Hold
D	3	Prog/Ramp Status (MSB)		see [2]
D	2	Prog/Ramp Status		see [2]
D	1	Prog/Ramp Status		see [2]
D	0	Prog/Ramp Status (LSB)		see [2]

[1] Segment number is a nibble with a value 1 to 8 corresponding to the currently active segment.
[2] Program or Ramp Status is a nibble having the value 0 to 6, the values mean:

D values and meaning

Value	Meaning	Access
0	Reset	R/W
1	N/A	
2	Run	R/W
3	Hold	R/W
4	Program end	R/O
5	Ramp End (Still active*)	R/O
6	Holdback Active	R/O

*After completing a ramp should PV deviate from SP1, the working set point will ramp back to SP1 at the current ramp rate.

3.9 Extension Status Word — XS

Digit	Bit	Function	Access	Clear/Set
A	15	Motor control (MSB)	R/W	See [3]
A	14	Motor control	R/W	See [3]
A	13	Motor control	R/W	See [3]
A	12	Motor control (LSB)	R/W	See [3]
B	11	Program No. (MSB)	R/W	See [4]
B	10	Program No.	R/W	See [4]
B	9	Program No.	R/W	See [4]
B	8	Program No. (LSB)	R/W	See [4]
C	7	Dig Output Channel 2	R/W	Off/On
C	6	Digital Input 3	R/O	Off/On
C	5	Select active PID	R/W	PID2/PID1
C	4	PID independent of SP	R/W	PID+SP/PID separate SP
D	3	N/A		
D	2	Enable Broadcast Mode	R/W	Disable/Enable
D	1	Adaptive Tune	R/W	/AT Active
D	0	Self Tune	R/W	/ST Active

[3] For valve positioner configurations only. Motor control is a nibble with a value 1 to 4 used to indicate or control the raise and lower outputs. In auto control digit A is read only and will return a value 0, 1 or 2. In manual digit A is read or write and values 3 and 4 are also valid to provide a nudge facility. A single write of value 3 will cause a lower pulse of minimum on-time to be presented to the motor, a write of 4 will cause a raise pulse for the minimum on-time. The values mean:

0	Raise and lower outputs off
1	Lower output active
2	Raise output active
3	Nudge lower
4	Nudge raise

[4] Program number is a nibble with a value 0 to 15 corresponding to the current program number. Range is configuration (controller model number) dependent. The values mean:

0	902P	R/O
1-4	903P	R/W
1-15	904P	R/W

Program number is also available more directly using the mnemonic CP.

3.10 Digital Output Status Word 1 — O1

Digit	Bit	Function	Attribute	Clear/Set
A	15	dwell 8 to channel 3	R/W	on/off[1]
A	14	ramp 8 to channel 3	R/W	on/off
A	13	dwell 7 to channel 3	R/W	on/off
A	12	ramp 7 to channel 3	R/W	on/off
B	11	dwell 6 to channel 3	R/W	on/off
B	10	ramp 6 to channel 3	R/W	on/off
B	9	dwell 5 to channel 3	R/W	on/off
B	8	ramp 5 to channel 3	R/W	on/off
C	7	dwell 4 to channel 3	R/W	on/off
C	6	ramp 4 to channel 3	R/W	on/off
C	5	dwell 3 to channel 3	R/W	on/off
C	4	ramp 3 to channel 3	R/W	on/off
D	3	dwell 2 to channel 3	R/W	on/off
D	2	ramp 2 to channel 3	R/W	on/off
D	1	dwell 1 to channel 3	R/W	on/off
D	0	ramp 1 to channel 3	R/W	on/off

3.11 Digital Output Status Word 2 — O2

Digit	Bit	Function	Attribute	Clear/Set
A	15	N/A		
A	14	N/A		
A	13	N/A		
A	12	N/A		
B	11	N/A		
B	10	N/A		
B	9	N/A		
B	8	N/A		
C	7	N/A		
C	6	N/A		
C	5	N/A		
C	4	N/A		
D	3	N/A		
D	2	N/A		
D	1	N/A		
D	0	END to channel 3	R/W	on/off

Bits 1 to 15 are for future expansion of channel 3 digital output.

[1] The bits in all the status words, 01 to 06, can be used to set digital outputs that are configured to be under program control; each bit controls the digital output during the specified program segment, so for example if this bit is set then the channel 3 digital output is on during the 8th dwell phase of the loaded program.

3.12 Digital Output Status Word 3 — O3

Digit	Bit	Function	Attribute	Clear/Set
A	15	dwell 8 to channel 4	R/W	on/off
A	14	ramp 8 to channel 4	R/W	on/off
A	13	dwell 7 to channel 4	R/W	on/off
A	12	ramp 7 to channel 4	R/W	on/off
B	11	dwell 6 to channel 4	R/W	on/off
B	10	ramp 6 to channel 4	R/W	on/off
B	9	dwell 5 to channel 4	R/W	on/off
B	8	ramp 5 to channel 4	R/W	on/off
C	7	dwell 4 to channel 4	R/W	on/off
C	6	ramp 4 to channel 4	R/W	on/off
C	5	dwell 3 to channel 4	R/W	on/off
C	4	ramp 3 to channel 4	R/W	on/off
D	3	dwell 2 to channel 4	R/W	on/off
D	2	ramp 2 to channel 4	R/W	on/off
D	1	dwell 1 to channel 4	R/W	on/off
D	0	ramp 1 to channel 4	R/W	on/off

3.13 Digital Output Status Word 4 — O4

Digit	Bit	Function	Attribute	Clear/Set
A	15	N/A		
A	14	N/A		
A	13	N/A		
A	12	N/A		
B	11	N/A		
B	10	N/A		
B	9	N/A		
B	8	N/A		
C	7	N/A		
C	6	N/A		
C	5	N/A		
C	4	N/A		
D	3	N/A		
D	2	N/A		
D	1	N/A		
D	0 END to channel 4	R/W	on/off	

Bits 1 to 15 are for future expansion of channel 4 digital output.

3.14 Digital Output Status Word 5 — O5

Digit	Bit	Function	Attribute	Clear/Set
A	15	dwell 8 to channel 2	R/W	on/off
A	14	ramp 8 to channel 2	R/W	on/off
A	13	dwell 7 to channel 2	R/W	on/off
A	12	ramp 7 to channel 2	R/W	on/off
B	11	dwell 6 to channel 2	R/W	on/off
B	10	ramp 6 to channel 2	R/W	on/off
B	9	dwell 5 to channel 2	R/W	on/off
B	8	ramp 5 to channel 2	R/W	on/off
C	7	dwell 5 to channel 2	R/W	on/off
C	6	ramp 4 to channel 2	R/W	on/off
C	5	dwell 3 to channel 2	R/W	on/off
C	4	ramp 3 to channel 2	R/W	on/off
D	3	dwell 2 to channel 2	R/W	on/off
D	2	ramp 2 to channel 2	R/W	on/off
D	1	dwell 1 to channel 2	R/W	on/off
D	0	ramp 1 to channel 2	R/W	on/off

3.15 Digital Output Status Word 6 — O6

Digit	Bit	Function	Attribute	Clear/Set
A	15	N/A		
A	14	N/A		
A	13	N/A		
A	12	N/A		
B	11	N/A		
B	10	N/A		
B	9	N/A		
B	8	N/A		
C	7	N/A		
C	6	N/A		
C	5	N/A		
C	4	N/A		
D	3	N/A		
D	2	N/A		
D	1	N/A		
D	0	END to channel 2	R/W	on/off

Bits 1 to 15 are for future expansion of channel 2 digital output.

4 900 EPC Simple Mnemonics

Mne	Meaning	Loop	Type
1H	PV high limit	Yes	Numeric
1L	PV low limit	Yes	Numeric
1M	PV measured value	Yes	Numeric
1T	Pulse burner 1 on time	No	Numeric
2M	pv auxilary measured value	Yes	Numeric
2R	ratio setpoint 2	No	Numeric
2T	Pulse burner 2 on time	No	Numeric
3M	remote input measured value	Yes	Numeric
3T	Pulse burner 3 on time	No	Numeric
4T	Pulse burner 4 on time	No	Numeric
5T	Pulse burner 5 on time	No	Numeric
6T	Pulse burner 6 on time	No	Numeric
7T	Pulse burner 7 on time	No	Numeric
8T	Pulse burner 8 on time	No	Numeric
A1	Alarm 1 setpoint	No	Numeric
A2	Alarm 2 setpoint	No	Numeric
A3	Alarm 3 setpoint	No	Numeric
A4	Alarm 4 setpoint	No	Numeric
AS	Current Autotune stage	Yes	Numeric
AT	D.R.A Trigger value	Yes	Numeric
B2	Channel 2 On/Off Deadband	Yes	Numeric
BL	Comms block length	No	Hex Status Word [1]
BO	On off sensor break power	Yes	Numeric
BP	Sensor break power	Yes	Numeric
BV	VP control sensor break power	Yes	Numeric
CC	Cycle time (channel 2)	Yes	Numeric
CH	Cycle time (channel 1)	Yes	Numeric
CJ	CJC value	Yes	Numeric
CM	Pulse burner Minimum cycle time	No	Numeric
CP	current program selection	Yes	Numeric
CS	current segment of program	Yes	Numeric
CU	Customer Process Units	Yes	Text
D1	Alarm delay 1	No	Numeric
D2	Alarm delay 2	No	Numeric
D3	Alarm delay 3	No	Numeric

[1] When used in the 900 EPC to read and write BL the resulting bit pattern is treated as two binary numbers (unsigned) which give the values of the block length for the 900 EPC and the supervisory machine. The first two hex characters used, give the block length for the supervisor machine talking to the 900 EPC, the last two are read only and give the value of the block length on the 900 EPC; this is always FF (255). The supervisor may set the first two bytes to any value greater than or equal to hex 10 (16) and less than or equal to FF (255). The default for BL is 80FF i.e. the supervisor has block length of 128.

Mne	Meaning	Loop	Type
D4	Alarm delay 4	No	Numeric
DB	Channel 1 on/off deadband	Yes	Numeric
DF	Detuning factor	Yes	Numeric
DL	Last DRA Tuning strategy	Yes	Numeric
DP	Derivative Output	Yes	Numeric
DS	Current DRA stage	Yes	0 allow settling 1 wait for trigger 2 find peak 1 3 find zero 1 4 find peak 2 5 find zero 2 6 find peak 3 7 find zero 3 8 find peak 4 9 find zero 4 10 find peak 5 11 end on zero 4 abort 12 end on peak 4 abort 13 end on peak 5 abort 14 end on peak 5 found 15 prepare update
DT	Process delay time	Yes	Numeric
EE	Comms Error code	No	Hex
ER	Error value (PV-SP)	Yes	Numeric
EP	Selected Program for edit	No	Numeric
ES	Selected segment for edit	No	Numeric
F1	Fraction of Lp 1 IP, drvd I/P inst	Yes	Numeric
F2	Fraction of Lp 2 IP, drvd I/P inst	Yes	Numeric
FD	Program power fail recovery definition	No	0 continue 1 reset program 2 servo and continue 3 test pv deviation 4 test time duration 5 hold test time
FF	Feed Forward value	Yes	Numeric
FT	Time last power fail occured	No	Numeric
H2	Setpoint 2 high limit	Yes	Numeric
HA	Ratio setpoint high limit	No	Numeric
HB	Cutback high value	Yes	Numeric
HC	Output1/output2 deadband	Yes	Numeric
HO	output channel 1 max power limit	Yes	Numeric
HS	Working setpoint high limit	Yes	Numeric
HT	Relative humidity wet bulb correction	No	Numeric
IF	Process Input Filter value	Yes	Numeric
II	Instrument identity	No	>9050 for all 900EPC
IM	Instrument mode	No	0 instrument in operational mode 1 instrument in standby mode 2 instrument in configuration mode
IO	Integral Output	Yes	Numeric
L2	Setpoint 2 low limit	Yes	Numeric
LA	Ratio setpoint low limit	No	Numeric

Mne	Meaning	Loop	Type
LB	Cutback low value	Yes	Numeric
LC	Load time constant	Yes	Numeric
LI	process input linearised val	Yes	Numeric
LO	output channel 2 max power limit	Yes	Numeric
LR	remote input linearised values	Yes	Numeric
LS	Working setpoint low limit	Yes	Numeric
LT	Local Setpoint Trim	Yes	Numeric
Lc	Program cycles value	No	Numeric
Lr	Program cycles remaining	No	Numeric
Ls	Sub-program cycles remaining	No	Numeric
MD	Maximum Tuning Disturbance	Yes	Numeric
MO	Channel 1 minimum	Yes	Numeric
MR	Manual reset	Yes	Numeric
MT	Minimum Response time	Yes	Numeric
NE	Normalised Error	Yes	Numeric
O1	Actual output level for output 1	Yes	Numeric
O2	Actual output level for output 2	Yes	Numeric
OI	Program overshoot inhibition	Yes	Numeric
OP	Target output level	Yes	Numeric
OR	Output power rate limit	Yes	Numeric
P2	level 2 password	No	Numeric
P3	level 3 password	No	Numeric
PC	configuration password	No	Numeric
PD	Program power fail recovery durations	No	Numeric
PE	Pyrometer emmissivity	Yes	Numeric
PF	Power Feedback value	Yes	Numeric
PM	Process input sensor trim	Yes	Numeric
PO	Process input offset	Yes	Numeric
PP	Process input sensor break position	Yes	Numeric
PR	Programmer current segment ramp rate	No	Numeric
PS	Program selected for load	Yes	Numeric
PT	Current segment type of program	Yes	0 end 1 ramp 2 dwell 3 step 4 call
PV	Process variable	Yes	Numeric
PW	Programmer current segment wkg sp	Yes	Numeric
RB	Ratio Bias	No	Numeric
RD	Remote Input Sensor Break value	Yes	Numeric
RF	Remote Input Filter value	Yes	Numeric
RG	Relative output 2 gain	Yes	Numeric
RH	Remote channel 1 max power limit	Yes	Numeric
RP	Remote input sensor break position	Yes	Numeric
RR	Working setpoint rate limit	Yes	Numeric
RS	Ratio Setpoint	No	Numeric
RT	Remote setpoint trim(Offset)	Yes	Numeric
RV	Remote, ch 1 max pwr lim man val	Yes	Numeric
RW	Working Ratio Setpoint	No	Numeric
S2	Setpoint 2	Yes	Numeric
SC	Control clock	No	Numeric
SD	Program servo duration	No	Numeric
SF	Setpoint Feedforward Trim	No	Numeric
SL	Setpoint 1	Yes	Numeric

Mne	Meaning	Loop	Type
SP	Working setpoint	Yes	Numeric
SR	Remote setpoint	Yes	Numeric
SS	Current segment of sub-program	Yes	Numeric
ST	Current segment type of sub-prog	Yes	Numeric
T1	Warm up title 1 screen	No	Text
T2	Warm up title 2 screen	No	Text
T3	Warm up title 3 screen	No	Text
T4	Warm up title 4 screen	No	Text
TD	Derivative time	Yes	Numeric
TG	Programmer current segment Target sp	No	Numeric
TI	Integral time	Yes	Numeric
TM	Time remaining in current segment	Yes	Numeric
TP	Valve output update time	Yes	Numeric
TR	Ratio Setpoint trim	No	Numeric
TS	Tuning sample time	Yes	Numeric
TT	motor travel time	Yes	Numeric
V0	Version number (68070 software)	No	Hex
V1	Version number (6805 software)	No	Hex
VO	Actual output for valve position	Yes	Numeric
WA	Status Word for Alarms	No	Hex
WI	Status Word for Digital Inputs	No	Hex
WL	Status Word Loop Status	Yes	Hex
WO	Status Word for Digital Outputs	No	Hex
WP	Status Word for Programmer	No	Hex
WS	Status Word for instrument	No	Hex
XP	Proportional band	Yes	Numeric
ZN	Current Ziegler Nichols stage	Yes	Numeric
br	Instrument baud rate	No	0 9600 baud 1 19200 baud 2 4800 baud 4 2400 baud 5 1200 baud 7 300 baud
ce	Clear error log	No	0 no action 1 clear error log
cj	CJC Calibration	No	Numeric
ck	Comms keys	No	Numeric
dh	Derived input high limit	Yes	Numeric
dl	Derived input low limit	Yes	Numeric
dt	Display test	No	0 no test 1 vertical odd 2 vertical even 3 horizontal odd 4 horizontal even 5 checker board odd 6 checker board even
ea	Error address	No	Numeric
et	telemetry enable (conf mode only)	No	Numeric
ev	Error vector	No	Numeric
l2	Level 2 access control	No	Numeric
la	Programmer loop access control	No	Numeric

Mne	Meaning	Loop	Type
ld	Program loading access	No	Numeric
lr	loop calibration request	Yes	0 calibration stopped 1 -20/+100 input voltage calibration 2 -10/+50 input voltage calibration 3 -20/+20 input voltage calibration 4 -8/+8 input voltage calibration 5 rt calibration with volts (measured externally) 6 rt calibration (volts measured internally) 7 high level input calibration 8 cjc calibration 9 perform low calibration 10 perform high calibration 11 copy loop calibrations from nonvol 12 confirm copy loop calibrations from nonvol 13 acknowledge loop calibration
ls	loop calibration status	Yes	0 not calibrating 1 ready 2 point calibrate 2 busy high calibration 3 busy low calibration 4 busy cjc calibration 5 calibration complete 6 ready to copy from nonvol 7 zero offset error 8 low calibration error 9 high calibration error 10 calibration input too noisy 11 external input out of range 12 hardware failure 13 input count zero 14 cjc input error
lv	loop calibration value	Yes	Numeric
md	Mains definition	No	0 50 Hz 1 60 Hz
mn	Display i/o slot number	No	1 display slot 1 2 display slot 2 3 display slot 3 4 display slot 4 5 display slot 5 6 display slot 6
mr	module calibration request	No	0 module calibration stopped 1 perform low input calibration 2 perform high input calibration 3 perform low output calibration 4 perform high output calibration 5 read value 6 acknowledge module calibration 7 recover module backup calibrations 8 recover all default calibrations 9 initialise all backup calibrations 10 confirm backup recovery 11 confirm default recovery 12 confirm backup initialisation

Mne	Meaning	Loop	Type
ms	module calibration status	No	0 module not calibrating 1 module output calibration 2 module input calibration 3 module output ready 4 module calibration complete 5 module zero offset error 6 module low calibration error 7 module high calibration error 8 module calibration input too noisy 9 module external input out of range 10 module hardware failure 11 module input count zero 12 module hardware not found 13 ready to get backup module calibrations 14 ready to get default calibrations 15 ready to initialise backup calibrations
mv	module calibration value	No	Numeric
na	Instrument node address	No	Numeric
nb	Pulse Burner No. of burners	No	Numeric
np	No. of process inputs	No	Numeric
pa	program modify access	No	Numeric
ra	Ratio access control	No	Numeric
rd	ratio type definition	No	0 ratio setpoint used as divisor 1 ratio setpoint used as multiplier
sp	Subprogram definition	No	Numeric
td	Timer configuration	No	Numeric
wt	Warm up screen type	No	Numeric
xa	Transducer access	No	Numeric

5 900 EPC Status Words

The 900 EPC uses 16 bit and 32 bit status words. Status words are transmitted using hex format and the '>' character (see section 2.4.5).

Status words in the 900 EPC have long latency read and writes. In particular writes are acknowledged by an ACK before they are implemented, i.e. when the message has been successfully received. The writer should not perform another write to a status word or composite until EE is clear, (see 2.10).

5.1 WA Status Word

	Function	Access
0	Historical Alarm 1 status	r/w
1	Historical Alarm 2 status	r/w
2	Historical Alarm 3 status	r/w
3	Historical Alarm 4 status	r/w
4-7	Not used	
8	Current Alarm 1 status	r/o
9	Current Alarm 2 status	r/o
10	Current Alarm 3 status	r/o
11	Current Alarm 4 status	r/o
12-15	Not Used	

Historical alarm status	
Value	Meaning
reading 1	indicates unacknowledged alarm exists
writing 0	acknowledges that alarm

Current alarm status	
Value	Meaning
reading 1	indicates currently in alarm
reading 0	indicates out of alarm

Latching alarms		
Historical	Current	Meaning
0	0	No alarm
0	1	Active alarm that has been acknowledged
1	0	Alarm no longer active not yet acknowledged
1	1	Active alarm not yet acknowledged

5.2 WS Status Word

Bit	Function	Access	Meaning
0	Keylock status	r/w	0=Keys 1=No Keys
1	Digital Lockout status	r/w	0=Dig Ips 1=No Dig Ips
2	Comms Disable status	r/w	0=Full Comms 1=Read Only Comms [1]
3	Comms Retransmission status		— (not yet implemented)
4	Comms Format status	r/w	0=ASCII 1=IEEE
5	Unacknowledged Alarm	r/o	0=No Alarm 1=Unack Alarm
6	Keys to be driven by comms	r/w	0=ON 1=OFF
7	Time Function status	r/w	0=Scheduler Off 1=Scheduler On
8 and 9	Clock Write Status	r/w	Bit 8 / Bit 9 / Meaning 0 0 Running 0 1 Set the Time 1 0 Clock stopped 1 1 Clock stopped
10	Not Used		
11	Not Used		
12	Not Used		
13	Not Used		
14 and 15	Instrument mode	r/o	Bit 15 / Bit 14 / Meaning 0 0 Operating mode 0 1 Standby mode 1 0 Configuration mode

[1] Once comms writes have been disabled they cannot be re-enabled over the comms link! The only way to re-enable comms writes is to use a digital input configured to disable comms. If this is closed and then re-opened manually comms writes will be re-enabled. (The digital input can then be re-used for something else.)

5.3 WI Status Word

Bit	Function	Access	Meaning	Slot
0	Digital Input 1 status	r/o	1=active 0=closed	1
1	Digital Input 2 status	r/o	1=active 0=closed	
2	Digital Input 3 status	r/o	1=active 0=closed	
3	Digital Input 4 status	r/o	1=active 0=closed	
4	Digital Input 5 status	r/o	1=active 0=closed	2
5	Digital Input 6 status	r/o	1=active 0=closed	
6	Digital Input 7 status	r/o	1=active 0=closed	
7	Digital Input 8 status	r/o	1=active 0=closed	
8	Digital Input 9 status	r/o	1=active 0=closed	3
9	Digital Input 10 status	r/o	1=active 0=closed	
10	Digital Input 11 status	r/o	1=active 0=closed	
11	Digital Input 12 status	r/o	1=active 0=closed	
12	Digital Input 13 status	r/o	1=active 0=closed	4
13	Digital Input 14 status	r/o	1=active 0=closed	
14	Digital Input 15 status	r/o	1=active 0=closed	
15	Digital Input 16 status	r/o	1=active 0=closed	
16	Digital Input 17 status	r/o	1=active 0=closed	5
17	Digital Input 18 status	r/o	1=active 0=closed	
18	Digital Input 19 status	r/o	1=active 0=closed	
19	Digital Input 20 status	r/o	1=active 0=closed	
20	Digital Input 21 status	r/o	1=active 0=closed	6
21	Digital Input 22 status	r/o	1=active 0=closed	
22	Digital Input 23 status	r/o	1=active 0=closed	
23	Digital Input 24 status	r/o	1=active 0=closed	
24	Digital Input 25 status	r/o	1=active 0=closed	Micro-board Dig in 1
25	Digital Input 26 status	r/o	1=active 0=closed	Micro-board Dig in 2
26-31	Not Used			

5.4 WO Status Word

Bit	Function	Access	Meaning	Slot
0	Digital Output 1 status	r/o	1=active	1
1	Digital Output 2 status	r/o	1=active	
2	Digital Output 3 status	r/o	1=active	
3	Digital Output 4 status	r/o	1=active	
4	Digital Output 5 status	r/o	1=active	2
5	Digital Output 6 status	r/o	1=active	
6	Digital Output 7 status	r/o	1=active	
7	Digital Output 8 status	r/o	1=active	
8	Digital Output 9 status	r/o	1=active	3
9	Digital Output 10 status	r/o	1=active	
10	Digital Output 11 status	r/o	1=active	
11	Digital Output 12 status	r/o	1=active	
12	Digital Output 13 status	r/o	1=active	4
13	Digital Output 14 status	r/o	1=active	
14	Digital Output 15 status	r/o	1=active	
15	Digital Output 16 status	r/o	1=active	
16	Digital Output 17 status	r/o	1=active	5
17	Digital Output 18 status	r/o	1=active	
18	Digital Output 19 status	r/o	1=active	
19	Digital Output 20 status	r/o	1=active	
20	Digital Output 21 status	r/o	1=active	6
21	Digital Output 22 status	r/o	1=active	
22	Digital Output 23 status	r/o	1=active	
23	Digital Output 24 status	r/o	1=active	
24-31	Not Used			

5.5 WP Status Word

Bit	Function	Access	Meaning
0	Not Used		
1	Not Used		
2	Power Fail while running	r/o	
3	Prog Reset after Power Fail	r/o	
4	Wait Until Loop 1	r/w	1=Wait
5	Wait Until Loop 2	r/w	1=Wait
6	Load Loop 1 Program Request	r/w	1=Load Program
7	Load Loop 2 Program Request	r/w	1=Load Program
8	Loop 1 Skip Segment	w/o	1=Skip, always reads as 0
9	Loop 2 Skip Segment	w/o	1=Skip, always reads as 0
10	Holdback Disable	r/w	0=Holdback 1=No holdback
11	Log Hold status	r/o	0=No hold 1=Hold has occurred
12	Log Holdback status	r/o	0=No holdback 1=Hdbk has occurred
13	Holdback Status	r/o	0=No holdback 1=Holdback
14 and 15	Programmer status	r/w	Bit 15 Bit 14 Meaning 0 1 Reset Program 1 0 Run Program 1 1 Hold Program

5.6 WL Status Word

Bit	Function	Access	Meaning
0	Auto/Manual status	r/w	0=Auto 1=Manual
1	Remote status	r/w	0=Local 1=Remote
2	Ratio Enabled status	r/w	0=Normal 1=Ratio
3	Autotune status	r/w	0=Off 1=Autotune
4	Adaptive tune status	r/w	0=Off 1=Adaptive Tune
5	Gain Scheduling status	r/w	0=Off 1=Gain Scheduling
6	Not Used		
7	Process Input status	r/o	0=Normal 1=Break
8	Remote Input status	r/o	0=Normal 1=Break
9	Setpoint 2 status	r/w	0=Off 1=Setpoint 2
10	Output Rate Limit status	r/w	0=Off 1=OP Limit
11	Setpoint Rate Limit status	r/w	0=Off 1=SP Limit
12	Cascade Status; Loop 1 only	r/w	0=Off 1=Cascade
13	Ratio Sp 2 Status Loop 1	r/w	0=Off 1=Ratio SP 2
14	Not Used		
15	Not Used		

6 900 EPC Composite Mnemonics

This sections describes all the 900 EPC composite mnemonics. The structure of the bisync message for each mnemonic is described. For example the alarm configuration mnemonic is described by the following text

```
ac(GS)Alarm 1(RS)Alarm 2(RS)Alarm 3(RS)Alarm 4
                     !
                     !
                     !
       .........!.....................................
       !             !            !             !              !
       !             !            !             !              !
    Alarm type   Parameter    Alarm         Latching       Delay Alarm
                 monitored    Hysterisis    Non-latching   Configuration
                 for Alarm    value         definition
```

This means that the ac mnemonic has data specified using a group separator GS, and the record separators RS. It is thus a two level structure, with four identical records, for each alarm, separated in the composite message by the record separator character RS. Each record has five fields in it separated by the field separator US.

The values of each field are given in a separate table, for example

Alarm type	
0	None
1	Full scale high
2	Full scale low
3	Deviation high
4	Deviation low
5	Deviation band
6	Maximum rate of change
7	Sensor break
8	Instrument fault

indicates that the alarm type field has 9 possible values, encoded as ascii 0-8.

If there is no table for a field, and nothing else is stated, it can be assumed that the field will contain a floating point number, encoded as described in section 2.4.5.

Unless otherwise stated the parameters do not require any channel identifier (see section 2.4.3).

Composites in the 900 EPC have long latency read and writes. In particular writes are acknowledged by an ACK before they are implemented, i.e. when the message has been successfully received. The writer should not perform another write to a status word or composite until EE is clear, (see section 2.10).

6.1 Tables of composite parameters

Mnemonic	Description	Section
	900 EPC User Configuration Mnemonics	
ac	Alarm Configuration	6.2.1
cc	Control configuration	6.2.2
st	Setpoint tracking	6.2.3
fc	Function configuration	6.2.4
tc	Tuning configuration	6.2.5
dc	Digital Comms configuration	6.2.6
oc	Output configuration	6.2.7
ip	Input n definition	6.2.8
iu	Instrument units	6.2.9
pc	Program configuration	6.2.10
ts	Transducer scaling configuration	6.2.11
md	Mains definition	6.2.12
LD	Custom linearisation display values (loop dependent)	6.2.13
LV	Custom linearisation input values (loop dependent)	6.2.14
	900 EPC Instrument Configuration Mnemonics	
it	Instrument type	6.3.1
pi	Process input configuration	6.3.2
ri	Remote Input configuration	6.3.3
lc	Loop control definition	6.3.4
id	I/O Slots occupancy definition	6.3.5
di	Digital input functions (microboard)	6.3.6
sf	io slot function definition	6.3.7
	900 EPC Calibration Parameters	
co	DC Output Calibration	6.4.1
cp	Resistance Thermometer Calibration	6.4.2
ct	Thermocouple Calibration Value	6.4.3
hl	High Level I/P Calibration	6.4.4
cr	Remote Input Calibration	6.4.5
	900 EPC Calibration Parameters	
GS	Gain Scheduling parameters	6.5.1
	900 EPC Telemetry Parameters	
AI	Analogue Telemetery Input	6.6.1
AO	Analogue Telemetery Output	6.6.2
DI	Digital Telemetery Input	6.6.3
DO	Digital Telemetery Output	6.6.4
	900 EPC Timer Function Parameters	
tm	Day and Time	6.7.1
DE	Standby mode entry time	6.7.2
DX	Standby mode exit time	6.7.3
	900 EPC Diagnostic Parameters	
IE	Instrument error log	6.8.1

900 EPC User Interface Customisation Parameters		
aa	Alarm Access	6.9.1
ba	Bar Control Access	6.9.2
ca	Program Control Access	6.9.3
da	Program Detail Access	6.9.4
l2	Level 2 Access	6.9.5
ld	Program Loading Access	6.9.6
pa	Program Modify Access	6.9.7
ra	Ratio Access	6.9.8
sa	Program Status Access	6.9.9
ta	Tuning Access	6.9.10
xa	Transducer Access	6.9.11
900 EPC Programmer Parameters		
PD	Programmer Power fail recovery duration.	6.10.1
SA	Programmer Segment	6.10.4
SA	Program End Segment	6.10.5
PG	Program Data	6.10.6
PA	Program Data	6.10.2

6.2 User configuration Parameters

6.2.1 Alarm Configuration — ac

```
ac(GS)Alarm 1(RS)Alarm 2(RS)Alarm 3(RS)Alarm 4(RS)
                   !
                   !
                   !
       ............!......................................
       !           !          !            !             !
       !           !          !            !             !
   Alarm type  Parameter   Alarm       Latching       Delay Alarm
               monitored   Hysterisis  Non-latching   Configuration
               for Alarm   value       definition
```

Alarm type	
0	None
1	Full scale high
2	Full scale low
3	Deviation high
4	Deviation low
5	Deviation band
6	Maximum rate of change
7	Sensor break
8	Instrument fault

Alarm monitored parameter	
0	Process Variable loop 1
1	Process Variable loop 2
2	Output loop 1
3	Output loop 2

Alarm latching definition	
0	nonlatching
1	latching

Alarm delay definition	
0	no alarm delay
1	alarm delay

6.2.2 Control configuration — cc

```
cc(GS)Loop 1 Control(RS)Loop 2 Control
      configuration     configuration
                   !
     ..............!................................
     !                      !                      !
Reverse/Direct          Derivative          Power feedback
Acting definition    Action definition       configuration
```

reverse acting definition	
0	Reverse acting control
1	Direct acting control

derivative action definition	
0	error
1	process variable

power feedback configuration	
0	No power feedback
1	Power feedback configured

6.2.3 Setpoint tracking — st

```
st(RS)Manual Loop 1(US)Manual Loop 2(US)Remote Loop 1(US)Remote Loop 2(US)
       Tracking          Tracking         Tracking         Tracking
       definition        definition       definition       definition

            Ratio(US)Cascade Tracking
            Tracking    definition
            definition
```

manual track definition	
0	No manual tracking
1	Manual tracking

remote tracking definition	
0	No remote tracking
1	Remote tracking

ratio tracking definition	
0	No Ratio Tracking
1	Ratio Tracking

cascade tracking definition	
0	No cascade tracking
1	cascade tracking

6.2.4 Function configuration — fc

```
fc(GS)Standby Mode(RS)Scheduler(RS)Loop 1 Function(RS)Loop 2 Function
      Availability    Availab't     Availability        Availability
                                        !
                                        !
     ..........................................................
     !             !        !           !           !              !
 Sp Rate lim  Man funcn  Rem func  Op Rate lim  SP feed forw    feed forw
```

standby mode availability	
0	Not available
1	Available

scheduler availability	
0	no scheduler available
1	daily scheduler

sp rate limit availability	
0	Not available
1	Available

manual function availability	
0	Not available
1	Available

remote function availability	
0	Not available
1	Available

op rate limit availability	
0	Not available
1	Available

sp feedforward availability	
0	not available
1	available

feedforward availability	
0	not available
1	available

6.2.5 Tuning configuration — tc

```
tc(GS)Loop 1 Tuning(RS)Loop 2 Tuning
    configuration    configuration
             !
             !
   .........!............................
   !                 !                 !
Autotune        Adaptive Tuning    Adaptive Tuning
availability       availability         type
```

autotune availability	
0	Not available
1	Available

adaptive tune availability	
0	Not available
1	Available

adaptive tuning type	
0	Continuous Adaptive Tuning
1	Working Setpoint Scheduling
2	Process Value Scheduling
3	Output Actual Level Scheduling

6.2.6 Digital Comms configuration — dc

```
dc(RS)Digital Comms Type
```

digital comms type	
0	no digital comms type
1	EI slave

6.2.7 Output configuration — oc

```
oc(GS)io slot 1(RS)..(RS)io slot 6(RS)loop1(RS)loop2(RS)anlgue(RS)..(RS)anlgue
      normal             normal       cooling  cooling    op 1        op 6
     inverted           inverted       type     type      conf        conf
    definition         definition      defn     defn
```

where `anlgue op X conf` is the analogue output configuration, a record whose format is shown below (the US separator is used to separate the fields):

```
                         analogue output conf
                                !
                                !
.......................................................................
!                  !                 !               !              !
analogue op        analogue op     analogue op    analogue op    analogue op
current or volts   high limit       low limit     op range low   op range high
```

normal/inverted definition	
0	normal
1	inverted

cooling type definition	
0	Linear cooling
1	Fan
2	Water
3	Oil

analogue o/p current or volts definition	
0	Volts
1	milliAmps

6.2.8 Input n definition — ip

```
ip(GS)proc ip 1(US)proc ip 2(RS)rem ip 1(US)rem ip 2(RS)
      filter       filter       filter       filter
      def          def          def          def


     cjc def 1(RS)cjc def 2(RS)normal(US)normal(RS)input
                ! 		  inv	      inv	 sampling
                ! 		  defn	      defn	 10 or 20
           ...........
          !           !
       CJC type    CJC value
```

process i/p filter definition	
0	No filtering
1	single pole exponation filter

remote i/p filter definition	
0	No filtering
1	single pole exponantion filter

cjc type	
0	no cjc
1	internal cjc
2	external cjc

normal/inverted definition	
0	normal
1	inverted

input sampling rate	
0	20 Hz
1	10 Hz

6.2.9 Instrument units — iu

```
iu(RS)Loop 1 process(US)Loop 2 process(US)Proportional(US)
      Input units       Input units          Band
                                             Unit type

Derivative(US)setpoint(US)output
 Integral     rate limit  rate limit
 Times Units    units       units
```

process input units	
0	none
1	deg celcius
2	deg fahrenheit
3	deg kelvin
4	acidity
5	relative humidity
6	custom unit[1]

proportional band units	
0	percentage of instrument span
1	Engineering Units

derivative integral times units	
0	seconds
1	minutes

rate limits units	
0	per minute
1	per hour

6.2.10 Program configuration — pc

```
pc(RS)Holdback Definition(US)Sub-program definition(US)
        Prog Display o/p Definition(US)Fast Run Def
```

holdback definition	
0	holdback unavailable
1	holdback available

sub-program definition	
0	no subprogram available
1	subprogram available

programmer digital o/p definition	
0	program digital output unavailable
1	program digital output available

fast run definition	
0	no fast run
1	fast run in standby
2	always fast run

6.2.11 Transducer scaling configuration — ts

```
ts(RS)Transducer scaling type loop 1(US)Transducer scaling type loop 2
```

transducer scaling type	
0	none
1	gain only transducer scaling
2	gain offset transducer scaling
3	shunt cal transducer scaling

[1] If custom units are selected the CU mnemonic allows the user to enter custom units

6.2.12 Mains definition — md

Note that this is not a composite, but a single selectable value.

mains definition	
0	50 Hz
1	60 Hz

6.2.13 Custom linearisation display values (loop dependent) — LD

```
LD(RS)Custom linearisation(US)Custom linearisation(US).(US)Custom linearisation
        display value 1          display value 2              display value 16
```

6.2.14 Custom linearisation input values (loop dependent) — LV

```
LV(RS)Custom linearisation(US)Custom linearisation(US).(US)Custom linearisation
        input value 1            input value 2                input value 16
```

6.3 Instrument configuration parameters

6.3.1 Instrument type — it

This is a singly selectable parameter (i.e. not actually a composite)

instrument type	
1	Single loop programmer
2	Single loop controller
3	Dual loop controller
4	Ratio controller
5	Ratio/Normal controller
6	Ratio independent controller
7	Derived input single loop controller
8	Derived input dual loop controller
9	Humidity controller
10	Temperature and Humidity controller
11	Switchover dual input single loop controller
12	Selectable input single loop controller
13	Maximum input single loop controller
14	Minimum input single loop controller
15	Cascade controller
16	Dual loop programmer
17	Ratio/Normal programmer
18	Ratio independent programmer
19	Derived input single loop programmer
20	Derived input dual loop programmer
21	Humidity programmer
22	Temperature and Humidity programmer
23	Switchover dual input single loop programmer
24	Selectable input single loop programmer
25	Maximum input single loop programmer
26	Minimum input single loop programmer
27	Cascade programmer

6.3.2 Process input configuration — pi

```
pi(GS)Process Input 1(US)Process Input 1(US)Display 1(US)characterisation
      Linearisation       range units       resolution       type 1
      definition
(US)Process Input 1(US)Process Input 1
      range max           range min
(RS)Process Input 2(US)Process Input 2(US)Display 2(US)characterisation
      Linearisation       range units     resolution       type 2
      definition
(US)Process Input 2(US)Process Input 2
      range max           range min
```

linearisation definition	
1	J thermocouple
2	L thermocouple
3	K thermocouple
4	T thermocouple
5	R thermocouple
6	S thermocouple
7	B thermocouple
8	W/W 26 Eng. thermocouple
9	W5W 26 Eng. thermocouple
10	E thermocouple
11	P10/40 RHS thermocouple
12	C thermocouple
13	R20/40 RH thermocouple
14	Platinell thermocouple
15	G2 W/WRe26% thermocouple
16	Ni/NI18%Moly thermocouple
17	Moly5%Re/Moly41%Re thermocouple
18	W3W 25 HER thermocouple
19	W5W 26 BIC thermocouple
20	N Nilsil thermocouple
21	Q004 pyrometer
22	Q003 pyrometer
23	RO 26 ORK 35-2-3 pyrometer
24	IVD1 pyrometer
25	DT1 pyrometer
26	RO23 pyrometer
27	FP/GP 10 pyrometer
28	FP/GP 11 pyrometer
29	FP/GP 12 pyrometer
30	FP/GP 20 pyrometer
31	FP/GP 21 pyrometer
32	RT100
33	JIS 100
34	Square Root
35	linear
36	customised without cjc
37	customised with cjc
38	customised with emissivity
39	characterised

process input units	
0	process input units mV
1	process input units V
2	process input units mA

characterisation type	
1	J thermocouple
2	L thermocouple
3	K thermocouple
4	T thermocouple
5	R thermocouple
6	S thermocouple
7	B thermocouple
8	W/W 26 Eng. thermocouple
9	W5W 26 Eng. thermocouple
10	E thermocouple
11	P10/40 RHS thermocouple
12	C thermocouple
13	R20/40 RH thermocouple
14	Platinell thermocouple
15	G2 W/WRe26% thermocouple
16	Ni/NI18%Moly thermocouple
17	Moly5%Re/Moly41%Re thermocouple
18	W3W 25 HER thermocouple
19	W5W 26 BIC thermocouple
20	N Nilsil thermocouple
21	Q004 pyrometer
22	Q003 pyrometer
23	RO 26 ORK 35-2-3 pyrometer
24	IVD1 pyrometer
25	DT1 pyrometer
26	RO23 pyrometer
27	FP/GP 10 pyrometer
28	FP/GP 11 pyrometer
29	FP/GP 12 pyrometer
30	FP/GP 20 pyrometer
31	FP/GP 21 pyrometer
32	RT100
33	JIS 100

6.3.3 Remote Input configuration — ri

```
ri(GS)Remote Input 1(US)Remote Input 1(US)Remote Input 1(US)Remote Input 1
      Linearisation      Range Units         range max          range min
      definition
(US)Remote Input 1(US)Remote Input 1(RS)Remote Input 2(US)Remote Input 2
    display range    display range      Linearisation      Range Units
       max              min             definition
(US)Remote Input 2(US)Remote Input 2(US)Remote Input 2(US)Remote Input 2
     range max         range min        display range     display range
                                           max               min
```

remote input linearisation	
1	J thermocouple
2	L thermocouple
3	K thermocouple
4	T thermocouple
5	R thermocouple
6	S thermocouple
7	B thermocouple
8	W/W 26 Eng. thermocouple
9	W5W 26 Eng. thermocouple
10	E thermocouple
11	P10/40 RHS thermocouple
12	C thermocouple
13	R20/40 RH thermocouple
14	Platinell thermocouple
15	G2 W/WRe26% thermocouple
16	Ni/NI18%Moly thermocouple
17	Moly5%Re/Moly41%Re thermocouple
18	W3W 25 HER thermocouple
19	W5W 26 BIC thermocouple
20	N Nilsil thermocouple
21	Q004 pyrometer
22	Q003 pyrometer
23	RO 26 ORK 35-2-3 pyrometer
24	IVD1 pyrometer
25	DT1 pyrometer
26	RO23 pyrometer
27	FP/GP 10 pyrometer
28	FP/GP 11 pyrometer
29	FP/GP 12 pyrometer
30	FP/GP 20 pyrometer
31	FP/GP 21 pyrometer
32	RT100
33	JIS 100
34	Square Root
35	linear
36	customised without cjc[1]

remote input range units	
1	remote input units V
2	remote input units mA

[1] Linearised using the custom linearisation tables entered using the LD and LV mnemonics

6.3.4 Loop control definition — lc

```
lc(RS)Loop 1 control definition(US)Loop 2 control definition
```

control definition	
0	PID channel 1, no channel 2
1	PID channel 1, PID channel 2
2	ON/OFF channel 1, ON/OFF channel 2
3	ON/OFF channel 1, no channel 2
4	VP channel 1, no channel 2
5	pulse burner channel 1

6.3.5 I/O Slots occupancy definition — id

```
id(RS)I/O Slot 1(US)I/O Slot 2(US)I/O Slot 3(US)I/O Slot 4
      definition    definition   definition    definition

          (US)I/O Slot 5(US)I/O Slot 6
          definition    definition
```

slot occupancy definitions	
0	io slot empty
1	Single Relay module
2	Single Logic module
3	Single Triac module
4	DC Control module
5	DC Input module
6	V.P. Pot. Position input module
7	Dual Relay module
8	Dual Relay with load sense module
9	Quad Logic Input module
10	Quad Logic Output module
11	Triple Logic Output module
12	Transducer Power Supply module
13	Digital Communication module
14	Dual Triac module
15	Dual Triac with load sense module
16	DC Retrans module
17	Dual Relay with output mutual exclusion
18	Dual Triac with output mutual exclusion
19	Invalid ident

6.3.6 Digital input functions (microboard) — di

```
di(RS)Digital Input 1(US)Digital Input 2
      function             function
```

digital input functions	
0	No digital input function
1	Auto/Manual Loop 1
2	Remote enable loop 1
3	Setpoint 2 enable loop 1
4	Setpoint Rate Limit enable loop 1
5	Freeze Integral action on loop 1
6	Output Rate Limit enable loop 1
7	Autotune enable loop 1
8	Adaptive tune enable loop 1
9	Gain Scheduling enable loop 1
10	Ratio enable
11	Ratio setpoint 2 enable
12	cascade enable
13	Auto/Manual loop 2
14	Remote enable loop 2
15	Setpoint 2 enable loop 2
16	Setpoint Rate Limit enable loop 2
17	Freeze Integral action on loop 2
18	Output Rate Limit enable loop 2
19	Autotune enable loop 2
20	Adaptive tune enable loop 2
21	Gain Scheduling enable loop 2
22	Autotune enable both loops
23	Adaptive tune enable both loops
24	Gain Scheduling enable both loops
25	Keylock enable
26	Use input 2
27	Digital comms disable
28	Digital Retransmission disable
29	Broadcast disable
30	Standby mode enable
31	Timer disable
32	Setpoint 1 loop 1
33	Setpoint 1 loop 2
34	Setpoint 1 both loops
35	Setpoint 2 both loops

digital input functions (cont)	
36	Remote enable both loops
37	Run
38	Reset
39	Hold
40	Run/Hold
41	Hold/Run
42	Holdback disable
43	loop 1 Skip Current segment
44	loop 2 Skip Current segment
45	Alarm acknowledge
46	Telemetry digital input
47	loop 1 l.s. digit of program no.
48	loop 1 2nd l.s. digit of program no.
49	loop 1 3rd l.s. digit of program no.
50	loop 1 m.s. digit of program no.
51	loop 2 l.s. digit of program no.
52	loop 2 2nd l.s. digit of program no.
53	loop 2 3rd l.s. digit of program no.
54	loop 2 m.s. digit of program no.
55	loop 1 wait until
56	loop 2 wait until
57	both loops wait until
58	load program loop 1
59	load program loop 2
60	load program both loops
61	Raise key
62	Lower key

6.3.7 io slot function definition — sf

The values read back shown in the table below depend on the io slot configured option, which may be found by reading the mnemonic id.

```
sf(GS)io slot 1(RS).............(RS)io slot 6
      function                      function
      definition                    definition
          !
     ...................................
     !          !           !           !
  io slot    io slot     io slot     io slot
   first     second       third      fourth
 function   function    function    function
```

digital input functions	
0	No digital input function
1	Auto/Manual Loop 1
2	Remote enable loop 1
3	Setpoint 2 enable loop 1
4	Setpoint Rate Limit enable loop 1
5	Freeze Integral action on loop 1
6	Output Rate Limit enable loop 1
7	Autotune enable loop 1
8	Adaptive tune enable loop 1
9	Gain Scheduling enable loop 1
10	Ratio enable
11	Ratio setpoint 2 enable
12	cascade enable
13	Auto/Manual loop 2
14	Remote enable loop 2
15	Setpoint 2 enable loop 2
16	Setpoint Rate Limit enable loop 2
17	Freeze Integral action on loop 2
18	Output Rate Limit enable loop 2
19	Autotune enable loop 2
20	Adaptive tune enable loop 2
21	Gain Scheduling enable loop 2
22	Autotune enable both loops
23	Adaptive tune enable both loops
24	Gain Scheduling enable both loops
25	Keylock enable
26	Use input 2
27	Digital comms disable
28	Digital Retransmission disable
29	Broadcast disable
30	Standby mode enable

digital input functions (cont)	
31	Timer disable
32	Setpoint 1 loop 1
33	Setpoint 1 loop 2
34	Setpoint 1 both loops
35	Setpoint 2 both loops
36	Remote enable both loops
37	Run
38	Reset
39	Hold
40	Run/Hold
41	Hold/Run
42	Holdback disable
43	loop 1 Skip Current segment
44	loop 2 Skip Current segment
45	Alarm acknowledge
46	Telemetry digital input
47	loop 1 l.s. digit of program no.
48	loop 1 2nd l.s. digit of program no.
49	loop 1 3rd l.s. digit of program no.
50	loop 1 m.s. digit of program no.
51	loop 2 l.s. digit of program no.
52	loop 2 2nd l.s. digit of program no.
53	loop 2 3rd l.s. digit of program no.
54	loop 2 m.s. digit of program no.
55	loop 1 wait until
56	loop 2 wait until
57	both loops wait until
58	load program loop 1
59	load program loop 2
60	load program both loops
61	Raise key
62	Lower key

digital output definitions (logic / relay / triac)	
63	programmer control digital output 1
64	programmer control digital output 2
65	programmer control digital output 3
66	programmer control digital output 4
67	programmer control digital output 5
68	programmer control digital output 6
69	programmer control digital output 7
70	programmer control digital output 8
71	programmer control digital output 9
72	programmer control digital output 10
73	programmer control digital output 11
74	programmer control digital output 12
75	Programmer run status
76	Programmer hold status
77	Programmer reset status
78	Programmer complete status
79	Programmer holdback status
80	Programmer log hold status
81	Programmer log holdback status
82	Alarm output 1
83	Alarm output 2
84	Alarm output 3
85	Alarm output 4
86	Any alarm output
87	auto/manual status on loop 1
88	auto/manual status on loop 2
89	setpoint 2 status on loop 1
90	setpoint 2 status on loop 2
91	loop 1 shunt cal
92	loop 2 shunt cal
93	pulsed burner output one
94	pulsed burner output two
95	pulsed burner output three
96	pulsed burner output four
97	pulsed burner output five
98	pulsed burner output six
99	pulsed burner output seven
100	pulsed burner output eight
101	No output function
102	Telemetry output
103	Loop 1 Channel 1 Control Output
104	Loop 1 Channel 2 Control Output
105	Loop 2 Channel 1 Control Output
106	Loop 2 Channel 2 Control Output

analogue outputs	
101	No output function
102	Telemetry output
103	Loop 1 Channel 1 Control Output
104	Loop 1 Channel 2 Control Output
105	Loop 2 Channel 1 Control Output
106	Loop 2 Channel 2 Control Output
107	loop 1 setpoint
108	loop 1 process variable
109	loop 1 error
110	loop 1 output
111	loop 1 input
112	loop 2 setpoint
113	loop 2 process variable
114	loop 2 error
115	loop 2 output
116	loop 2 input

analogue inputs	
117	Loop 1 Remote Setpoint
118	Loop 1 Remote Setpoint Trim
119	Loop 1 process feedforward
120	Loop 1 Remote Channel 1 Power Limit
121	Loop 1 Remote Channel 1 Power Limit / Power Level
122	Ratio Setpoint Trim function
123	Process variable 3
124	Loop 2 Remote Setpoint
125	Loop 2 Remote Setpoint Trim
126	Loop 2 process feedforward
127	Loop 2 Remote Channel 1 Power Limit
128	Loop 2 Remote Channel 1 Power Limit / Power Level
129	No analogue input function
130	Telemetry analogue input

valve positioner relay	
103	Loop 1 Channel 1 Control Output
104	Loop 1 Channel 2 Control Output
105	Loop 2 Channel 1 Control Output
106	Loop 2 Channel 2 Control Output

6.4 Calibration Parameters

6.4.1 DC Output Calibration — co

```
co(GS)DC o/p low calibration(US)...DC o/p low calibration(RS)
               Calib number 1             Calib number 6

DC o/p high calibration(US)...DC o/p high calibration
            Calib number 1    Calib number 6
```

6.4.2 Resistance Thermometer Calibration — cp

This is loop dependent, so a channel identifier of 1 or 2 should be used.

```
cp(RS)RT Voltage Cal(US)RT Voltage Cal(US)RT Voltage Cal(US)RT Voltage Cal
         Low Point           High Point           Low Value            High Value
```

6.4.3 Thermocouple Calibration Value — ct

```
ct(GS)TC Low i/p Cal Value(RS)TC High i/p Cal Value
              !                              !
              !                              !
              !                              !
       .......!.........              .......!................
       !               !              !                      !
       !               !              !                      !
   TC i/p low........TC i/p low  TC i/p high...........TC i/p high
   calibration      calibration  calibration            calibration
     value 1          value 4    value1               value 4
```

6.4.4 High Level I/P Calibration — hl

This is loop dependent, so a channel identifier of 1 or 2 should be used.

```
hl(RS)High Level I/P Low Cal Value(US)High Level I/P High Cal Value
```

6.4.5 Remote Input Calibration — cr

This is loop dependent, so a channel identifier of 1 or 2 should be used.

```
cr(RS)Remote i/p low cal(US)Remote i/p high cal
```

This is loop dependent, so a channel identifier of 1 or 2 should be used.

6.5 Gain Scheduling Parameters

6.5.1 Gain Scheduling parameters — GS

This is loop dependent, so a channel identifier of 1 or 2 should be used.

```
GS(GS)Scheduling(RS)Rescheduling(RS)Scheduling .... Rescheduling(RS)Scheduling
       Control        boundary         Control         boundary         Control
        set 1            1              set 2             4              set 5
          !
          !
          !
    ......!...........................................................
    !              !              !            !             !            !
    !              !              !            !             !            !
Cutback High   Proportional   Integral   Derivative   Cutback low     Relative
                   band          time         time                    Op 2 gain
```

6.6 Telemetery Parameters

6.6.1 Analogue Telemetery Input — AI

```
AI(RS)input 1(US)input 2(US)...(US)input 8
```

6.6.2 Analogue Telemetry Output — AO

```
AO(RS)output 1(US)output 2(US)...(US)output 6
```

6.6.3 Digital Telemetry Input — DI

```
DI(RS)input 1(US)input 2(US)...(US)input 8
```

6.6.4 Digital Telemetry Output — DO

```
DO(RS)output 1(US)output 2(US)...(US)output 6
```

6.7 Timer Function Parameters

All times are encoded as integers. Days may have the values 1 to 7, hours values 1 to 23, minutes and seconds 1 to 59.

6.7.1 Day and Time — tm

```
tm(RS)day(US)hour(US)Minutes(US)Seconds
```

6.7.2 Standby mode entry time — DE

```
DE(RS)Entry Hour(US)Entry minute
```

6.7.3 Standby mode exit time — DX

```
DX(RS)Exit Hour(US)Exit minute
```

6.8 Diagnostics Parameters

6.8.1 Instrument error log — IE

```
IE(RS)error log item 1(US)error log item 2(US).........(US)error log item 16
```

error log definitions	
0	No log error
1	Watchdog error
2	Software watchdog error
3	Uninitialised interrupt error
4	Configuration data checksum error
5	Nonvolatile memory read access error
6	Nonvolatile memory write access error
7	Nonvolatile memory test pattern corruption
8	RAM test pattern corruption error
9	Stack error
10	ASIC 1 error
11	RTC error
12	SPI bus failure
13	50ms tasks too slow
14	calibration value out of limits
15	I2C bus failure
16	low battery
17	vp position pot oc loop 1 (open circuit - reads 127)
18	vp position pot sc loop 1 (short circuit - reads 0)
19	vp position pot oc loop 2 (open circuit - reads 127)
20	vp position pot sc loop 2 (short circuit - reads 0)
21	pv input overrange loop 1 (sensor break)
22	pv input underrange loop 1 (sensor break)
23	pv input eids not ready loop 1 (reset inst)
24	pv input 6805 not ready loop 1 (reset inst)
25	pv input overrange loop 2 (sensor break)

error log definitions (cont)	
26	pv input underrange loop 2 (sensor break)
27	pv input eids not ready loop 2 (reset inst)
28	pv input 6805 not ready loop 2 (reset inst)
29	remote input overrange loop 1 (sensor break)
30	remote input underrange loop 1 (sensor break)
31	remote input eids not ready loop 1 (sensor break)
32	remote input 6805 not ready loop 1 (sensor break)
33	remote input overrange loop 2 (sensor break)
34	remote input underrange loop 2 (sensor break)
35	remote input eids not ready loop 2 (sensor break)
36	remote input 6805 not ready loop 2 (sensor break)
37	pv input eids selftest failure loop 1
38	pv input timer selftest failure loop 1
39	pv input rom selftest failure loop 1
40	pv input eids selftest failure loop 2
41	pv input timer selftest failure loop 2
42	pv input rom selftest failure loop 2
43	remote input eids selftest failure loop 1
44	remote input timer selftest failure loop 1
45	remote input rom selftest failure loop 1
46	remote input eids selftest failure loop 2
47	remote input timer selftest failure loop 2
48	remote input rom selftest failure loop 2

6.9 User Interface Customisation Parameters

6.9.1 Alarm Access — aa

```
aa(RS)Level 1 Alarm Access(US)Level 2 Alarm Access
```

alarm access	
0	access
1	hide

6.9.2 Bar Control Access — ba

```
ba(RS)Bar Control Access Loop 1(US)Bar Control Access Loop 2
```

bar control page access	
0	access
1	hide

6.9.3 Program Control Access — ca

```
ca(RS)Level 1 Program Control Access(US)Level 2 Program Control Access
```

program control access	
0	access
1	hide

6.9.4 Program Detail Access — da

```
da(RS)Level 1 Program Detail Access(US)Level 2 Program Detail Access
```

program detail access	
0	access
1	hide

6.9.5 Level 2 Access — l2

```
l2(RS)Control Page Access(US)Miscellaneous Page Access(US)Limits Page Access
```

level 2 access	
0	access
1	hide

6.9.6 Program Loading Access — ld

```
ld(RS)Level 1 Loading Access(US)Level 2 Loading Access
```

program loading access	
0	access
1	hide

6.9.7 Program Modify Access — pa

```
pa(RS)Level 1 Program Modify Access(US)Level 2 Program Modify Access
```

program modify access	
0	access
1	hide

6.9.8 Ratio Access — ra

```
ra(RS)Level 1 Ratio Access(US)Level 2 Ratio Access
```

ratio access definition	
0	access
1	hide

6.9.9 Program Status Access — sa

sa(RS)Level 1 Program Status Access(US)Level 2 Program Status Access

program status access	
0	access
1	hide

6.9.10 Tuning Access — ta

```
ta(RS)Level 1 Tuning Access(US)Level 2 Tuning Access
```

tuning access	
0	access
1	hide

6.9.11 Transducer Access — xa

```
xa(RS)Level 1 Transducer Access(US)Level 2 Transducer Access
```

transducer access	
0	access
1	hide

6.10 Programmer Parameters

6.10.1 Programmer Power fail recovery duration. — PD

Here durations are floating point seconds.

```
PD(RS)Powerfail servo duration(US)Powerfail reset duration
```

6.10.2 Program Data — PA

The PA mnemonic is used to read or write to the 900EPC program that has been selected using the mnemonic EP.

A full 900 EPC program consists of the program's general data, followed by N - 1 segment data sections, where N is the number of segments in the program including the end segment, and finally the end segment data. This may be read or downloaded in one composite, using mnemonic PA and channel number 3. The format of this mnemonic is complex, and it is described using the following "meta" characters `[] {} ()`. These do not appear in the bisync message!

The characters `[]` surround a set of options, separated by the character `|`, and the characters `{}` mean that there are one or more repeats of the enclosed data. The characters `()` surround items that must occur together, though as usual they are also used to surround message delimeter characters such as RS. So for example `[AAA|BBB]` means that the message may contain either AAA or BBB, and `{ AAAA(US)BBBB}` means that the sequence AAAA(US)BBBB may appear one or more times in the message.

The format of the PA composite data is shown below

```
3PA(GS)Program Type(RS)No of Segs in Prog(RS)Servo Start Selection(RS)
Prog Holdback Type(RS)Prog Holdback Val(RS)Dwell Time Unit(RS)
Ramp Time Unit(RS)Ramp Rate Resolution(RS)Ramp Type Def(RS)Target SP Resolution
 { (RS)Seg Type(US)Target SP(US)Dig O/p States(US)Dig O/P Change State
   (US)[ Ramp Rate | Time To target | Dwell duration |
             ( Sub-program No(US)No of Calls )]}
(RS)Seg Type(US)Program End Type(US)Dig O/p States(US)Dig O/P Change State(US)
     End Level Data(US)End Time Data
```

The first item after the PA mnemonic and channel number is the program type, viz:

program type	
0	Program
1	Subprogram

This is followed by the number of program segments, which will denote the number of repeats of the composite data surrounded by the `{}` characters. The subsequent fields to the first segment type field are described in the tables below.

servo start selection	
0	Servo from PV
1	Servo from SP

prog holdback type	
0	none
1	band deviation
2	high deviation
3	low deviation

dwell time units	
0	seconds
1	minutes
2	hours

ramp time units	
0	seconds
1	minutes
2	hours

ramp resolution	
0	0 decimal places
1	1 decimal places
2	2 decimal places
3	3 decimal places

ramp type def	
0	Ramp rate
1	Time to target

target setpoint resolution	
0	0 decimal places
1	1 decimal places
2	2 decimal places
3	3 decimal places

After these fields (i.e. directly after the Target SP resolution field) there will appear N segment records (including the end segment), where N is the second record in the whole of the message. A segment may have various types, viz:

Segment type		Optional associated data
0	end	No associated data, this type only appears in final segment record
1	ramp	Ramp rate or time to target is specified according to program ramp type def
2	dwell	Dwell duration is specified
3	step	No associated data
4	call	Sub program number and number of times it is called specified

This is recorded in the segment type field. Following this is the target setpoint data.

The next two fields that appear are the digital output states and digital output change states. The digital output states and digital output change state are status words, however they are encoded as integers whose binary bit values gives the values of the 12 digitals; the integers therefore may have values of between 0 to 4095. If a bit is set to zero in the digital output change state it means that this digital output is controlled by the other loop, and any value specified in the digital output state field is ignored. If a bit is one then the corresponding value in the same bit in the digital output state field is set, 1 for turning the digital output on, 0 for turning it off.

The data in the next field, after the digital output change state, depends on the segment type, as shown in the previous table. If the segment is a ramp then the field will either contain the floating point time to target or ramp rate according to the program's ramp type definition. If the segment type was dwell then the duration of the dwell is specified in floating point number of seconds. If the segment type is call then the sub-program to call is specified along with the number of times to call it. If the segment type is step then this field will be blank.

The N segment records are followed by the end segment which is denoted by reading a segment type of end. The end type of the program has values shown in the table that follows:

program end type	
0	indefinite dwell
1	ramp at rate to target
2	ramp for time to target
3	set output power
4	reset with setpoint track
5	reset without setpoint track
6	subprogram end

The digital output states and change states are as before.

If the program end type is ramp at rate or ramp for time then the program ramps to the set point level specified by the End Level Data at the rate or with the duration specified by the End Time Data.

If the program end type is set output power then the end level data specifies the set point applied.

Otherwise the End Level Data and End Time Data fields are left blank.

A Program end type of 6 (subprogram end) must be used for a sub-program.

6.10.3 900EPC Program Example

The following is an example of downloading and executing a 900EPC program. Only the data transmitted is shown, the bisync control characters are omitted for clarity.

The WP status word (section 5.5) is used to control loading and running of programs.

Reset the program currently loaded

```
SUPERVISOR :        WP>4000
INSTRUMENT :        <ACK>
```

Remove the programs loaded to each loop.

```
SUPERVISOR :        1PS 0.
INSTRUMENT :        <ACK>

SUPERVISOR :        2PS 0.
INSTRUMENT :        <ACK>

SUPERVISOR :        WP>00C0
INSTRUMENT :        <ACK>
```

Loading program 0 is equivalent to removing a program. It is necessary to first remove a program from the loop before editing it.

Select the program to edit

```
SUPERVISOR :        EP 1.
INSTRUMENT :        <ACK>
```

Transfer the program to the instrument

```
SUPERVISOR :

3PA(GS)0.(RS)5.(RS)0.(RS)2.(RS)5.(RS)0.(RS)0.(RS)0.(RS)1.(RS)0.
    (RS)3.(US)0.(US)0.(US)4095.(US)
    (RS)2.(US)(US)0.(US)4095.(US)60.
    (RS)1.(US)50.(US)4095.(US)4095.(US)60.
    (RS)4.(US)(US)0.(US)0.(US)11.(US)4.
    (RS)0.(US)2.(US)0.(US)4095.(US)0.(US)60.

INSTRUMENT :        <ACK>
```

The fields in the above message are described in detail below. The data is transmitted in either a single 128 character string or using the Ei-Bisynch block mode in smaller strings if the supervisor is not capable of holding the entire string. Note that because the data is a composite receiving the acknowledge merely indicates that the data has been received correctly (i.e. that the block sum check has passed), and not that the composite parses correctly.

Poll the Communications Error Status to check for successful write

```
SUPERVISOR :        EE
INSTRUMENT :        EE>0000
```

A response of hex 0000 indicates the program was validated. A response of hex 0046 indicates the program has not yet been validated and EE should be polled again until the response changes. Any other response indicates a fault in the program, see section 2.10 for details. Note that the validation is partial, further validation occurs on program load.

Select the program to load to the desired LOOP in the instrument

```
SUPERVISOR :        1PS 1.
INSTRUMENT :        <ACK>
```

The program was transferred from the supervisor to program 1 in the instrument. Now program 1 has been selected ready for loading to loop 1. Transmitting `2PS 1.` would select program 1 for loading to loop 2.

Load the selected program to the desired LOOP

```
SUPERVISOR :        WP>0040
INSTRUMENT :        <ACK>
```

The program is now loading to loop 1. The `<ACK>` reply does not indicate a successful load, but only that the command was received.

Poll the Communications Error Status to check for successful load

```
SUPERVISOR :        EE
INSTRUMENT :        EE>0000
```

A response of 0000 indicates a successful load. A response of 0046 indicates the load has not yet completed, EE should be polled again until the response changes. Any other response indicates a fault in the load, (see section 2.10 for details). Though the program data has been partially checked when the program was written, other errors are possible on load, for example a call to a non-existent sub-program.

Run the program

```
SUPERVISOR :        WP>8000
INSTRUMENT :        <ACK>
```

The program data would be transmitted in a message like

```
3PA(GS)0.(RS)5.(RS)0.(RS)2.(RS)5.(RS)0.(RS)0.(RS)0.(RS)1.(RS)0.
+ [4 * SEGMENTS]
+ [END SEGMENT]
```

Each of these fields is described in the table below:

Field separator and value	Comment
GENERAL DATA	
(GS) 0.	Program
(RS) 5.	5 Segments (including end segment)
(RS) 0.	Servo from PV on RUN
(RS) 2.	Deviation High Holdback
(RS) 5.	Holdback value = 5
(RS) 0.	Dwell times in Seconds
(RS) 0.	Ramp times in Seconds
(RS) 0.	Ramp resolution has 0 decimal places
(RS) 1.	Ramp as Time to Target
(RS) 0.	Target Setpoint resolution has 0 decimal places
SEGMENT 1	
(RS) 3.	Step Segment
(US) 0	Target setpoint = 0
(US) 0.	Digital outputs all OFF.
(US) 4095.	
(US)	Field not required for Step (OPTIONAL SEPARATOR)
SEGMENT 2	
(RS) 2.	Dwell Segment
(US)	Target setpoint not required (SEPARATOR COMPULSORY)
(US) 0.	Digital outputs all OFF.
(US) 4095.	
(US) 60.	Dwell for 60 seconds
SEGMENT 3	
(RS) 1.	Ramp Segment
(US) 50.	Target setpoint = 50
(US) 4095.	Digital outputs all ON.
(US) 4095.	
(US) 60.	Ramp Duration = 60 seconds
SEGMENT 4	
(RS) 4.	Sub Program Call
(US)	Target setpoint not required (SEPARATOR COMPULSORY)
(US) 0.	
(US) 0.	Digital outputs unchanged.
(US) 11.	Jump to SUB-Program 11.
(US) 4.	Cycle the Sub-Program 4 times.
END SEGMENT	
(RS) 0	End segment
(US) 2.	Time to Target ramp
(US) 0.	Digital Outputs all OFF.
(US) 4095.	
(US) 0	Target setpoint = 0
(US) 60.	Ramp Duration = 60 seconds.

Sub-program 11 would be created by writing 11 to EP. An example of a composite sub-program is shown below.

SUB-PROGRAM	
GENERAL DATA	
(GS) 1.	Sub-program
(RS) 4.	4 segments
(RS)	Servo type not required (SEPARATOR COMPULSORY)
(RS) 1.	Deviation Band Holdback
(RS) 15.	Holdback value = 15
(RS) 1.	Dwell times inMminutes.
(RS) 2.	Ramp times in Hours
(RS) 2.	Ramp resolution = 2 decimal places
(RS) 0.	Ramp type = Ramp rate.
(RS) 1.	Target setpoint resolution = 1 decimal place
SEGMENT 1	
(RS) 2.	Dwell segment
(US)	Target Setpoint not required (SEPARATOR COMPULSORY)
(US) 0.	Digital outputs all OFF.
(US) 4095.	
(US) 0.1	Dwell for 0.1 minutes
SEGMENT 2	
(RS) 3.	Step segment
(US) 33.3	Target setpoint = 33.3
(US) 0.	Digital outputs all OFF.
(US) 4095.	
(US)	Field not required for Step (OPTIONAL SEPARATOR)
SEGMENT 3	
(RS) 1.	Ramp Segment
(US) 66.6	Target Setpoint = 66.6
(US) 0.	Digital outputs all OFF.
(US) 4095.	
(US) 3.33	Ramp at 3.33 units/hour
END SEGMENT	
(RS) 0	End Segment
(US) 6.	Sub program end
(US) 0	Digital Outputs all OFF.
(US) 4095.	
(US)	Field not required for sub program end
(US)	(OPTIONAL SEPARATOR)

6.10.4 Programmer Segment — SA

Individual segments can be selected and read or written (i.e. edited) using the edit segment mnemonic (ES). This allows individual segment composite data to be read or written. The format is the same as the segment part of the program data PA mnemonic. Channel identifier 3 must be used.

```
3SA(RS)Seg Type(US)Target SP(US)Dig O/p States(US)Dig O/P Change State
(US)[Ramp Rate | Time To target | Dwell duration |(Sub-prog No(US)No of Calls)]
```

segment type	
1	ramp
2	dwell
3	step
4	call

6.10.5 Program End Segment — SA

If the last segment is selected by writing the last segment number to the ES mnemonic, then the SA composite will be read with the format of the last segment record in the PA mnemonic, viz:

```
3SA(RS)Seg Type(US)End Type(US)Dig O/p States(US)
Dig O/P Change State(US)End Level Data(US) End Time Data
```

Channel identifier 3 must be used.

The segment type will always be end (value 0).

6.10.6 Program Data — PG

The PG mnemonic can be used to read or write the program data part of the PA before any segment records.

```
3PG(GS)Program Type(RS)No of Segs in Prog(RS)Servo Start Selection(RS)
Prog Holdback Type(RS)Prog Holdback Val(RS)Dwell Time Unit(RS)
Ramp Time Unit(RS)Ramp Rate Resolution(RS)Ramp Type Def(RS)Target SP Resolution
```

Note that for compatibility with PA even though only two levels are used in this composite, GS and RS are used as separators.

Channel identifier 3 must be used.

INTERNATIONAL SALES AND SERVICE

AUSTRALIA
Eurotherm Pty. Ltd.
Telephone Sydney (+61 2) 94777022
Fax (+61 2) 94777756

AUSTRIA
Eurotherm GmbH
Telephone Vienna (+43 1) 7987601
Fax (+43 1) 7987605

BELGIUM
Eurotherm B.V.
Telephone Antwerp (+32 3) 3223870
Fax (+32 3) 3217363

DENMARK
Eurotherm A/S
Telephone Copenhagen (+45 31) 871622
Fax (+45 31) 872124

FRANCE
Eurotherm Automation SA
Telephone Lyon (+33 478) 664500
Fax (+33 478) 352490

GERMANY
Eurotherm Regler GmbH
Telephone Limburg (+49 6431) 2980
Fax (+49 6431) 298119
Also regional offices

HONG KONG
Eurotherm Limited
Telephone Hong Kong (+852) 28733826
Fax (+852) 28700148
Telex 0802 69257 EIFEL HX

INDIA
Eurotherm India Limited
Telephone Chennai (+9144) 4961129
Fax (+9144) 4961831

IRELAND
Eurotherm Ireland Limited
Telephone Naas (+353 45) 879937
Fax (+353 45) 875123

ITALY
Eurotherm SpA
Telephone Como (+39 31) 975111
Fax (+39 31) 977512
Telex 380893 EUROTH I

JAPAN
Eurotherm Japan Limited
Telephone Tokyo (+81 3) 33702951
Fax (+81 3) 33702960

KOREA
Eurotherm Korea Limited
Telephone Seoul (+82 2) 5438507
Fax (+82 2) 545 9758
Telex EIKOR K23105

NETHERLANDS
Eurotherm B.V.
Telephone Alphen aan den Rijn (+31 172) 411752
Fax (+31 172) 417260

NEW ZEALAND
Eurotherm International (NZ) Limited
Telephone Auckland (+64 9) 2635900
Fax: (+64 9) 2635901

NORWAY
Eurotherm A/S
Telephone Oslo (+47 66) 803330
Fax (+47 66) 803331

SPAIN
Eurotherm España SA
Telephone (+34 91) 6616001
Fax (+34 91) 6619093

SWEDEN
Eurotherm AB
Telephone Malmo (+46 40) 384500
Fax (+46 40) 384545

SWITZERLAND
Eurotherm Produkte (Schweiz) AG
Telephone Zurich (+41 55) 4154400
Fax (+41 55) 4154415

UNITED KINGDOM
Eurotherm Controls Limited
Telephone Worthing (+44 1903) 269888
Fax (+44 1903) 269666

U.S.A
Eurotherm Controls Inc.
Telephone Reston (+1 703) 7873405
Fax (+1 703) 7873436

ED 11

http://www.eurotherm.co.uk

Part No. HA023776

Printed in England