800

SERIES

EUROTHERM
CONTROLS

Communications Handbook

800 SERIES INSTRUMENT COMMUNICATIONS

Contents

Page

1.0 GENERAL DESCRIPTION

Eurotherm 800 series communicating controllers, in accordance with RS422 specification can provide two way asynchronous serial communication with a computer; the protocol used corresponds to ANSI X3.28 rev. '76 subcategories 2.5 and A4. This protocol defines the necessary procedures to perform the communication functions of establishing connection, message transference, and termination of the connection. Messages consist of ASCII characters including non-printing "control" characters and can be classified in two sequences, one to read information from the 800 (Section 2) and one to send data to the 800 instrument(Section 3).
This information is valid for 808, 815, 818, 820, 821, 822, 825 and 847 instruments and Sections 1 to 3 apply to all of them. Sections 4 to 7 list the specific details relevant to the different controllers.

1.1 Formal Specification

1.1.1 Multi Drop Supervisory Link

Transmission Standard	:RS485 (RS422)(bi-directional)
Protocol	:ANSI-X3.28-2.5-A4
Data Rates	300,600,1200,2400,3600,4800 or 9600 baud
Character Format (300 to 9600 baud)	:ASCII + 1 start, 1 parity and 1 stop bit.
Parity	:Even

Note: The 820 series supports a baud rate of 110 with 2 stop bits.

1.1.2 Single Serial Link

Transmission Standard	:RS-232 (bi-directional)
Protocol	:ANSI-X3.28-2.5-A4
Data Rates	:300,600,1200,2400,3600,
Character Format (300 to 9600 baud)	:ASCII + 1 start, 1 parity and 1 stop bit.
Parity	:Even

Note: The 820 series supports a baud rate of 110 with 2 stop bits.

1.1.3 Digital Communications

	RS232	RS422	RS485
Electrical Connections	3-wire, single ended	4-wire, differential	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	

1.2 Explanation of Terms

1.2.1 Address

The 800 has an address of two digits, the first a 'group' number 0 to 9 and the second a 'unit' number 0 to 9. There are, therefore, 100 different addresses 00 to 99.

1.2.2 Mnemonics

Information is exchanged with an 800 using mnemonics of two characters to specify the actual parameter - for example

PV is the process variable
XP is the proportional band etc.
Full tables are available for each type of instrument in Sections 4 to 7.

1.2.3 Control Characters

Six non-printing ASCII control characters are used to control the messages;

ASCII-HEX		
02	(STX)	Start of Text
03	(ETX)	End of Text
04	(EOT)	End of Transmission
05	(ENQ)	Enquiry
06	(ACK)	Positive Acknowledge
15	(NAK)	Negative Acknowledge

1.2.4 Data Formats

Associated with each mnemonic (parameter) is a value for which there are two possible data formats, Free and Fixed.

i) Free. This is the format normally used.

There are five or six characters used it is six characters in the 818 and the 820 series with the five digit option). These five or six characters include the number and if necessary the decimal point and minus sign. Leading or trailing zeros or leading spaces may be used to pad out the data, eg:

13.9 may be sent as 013.9
13.9
13.9 etc.

The minus sign must precede the value, eg:

-2 may be sent as -2
-002.
-0002 etc.

The decimal point and the minus sign are treated as one complete character. When returning the data the 800 will always place the decimal point (or the minus sign) in the same place as it appears on the display on the controller and it will pad it out with spaces up to the 5 (or 6) characters.

ii Fixed Format (This corresponds to the TCS ASCII Protocol).
Spaces are not permitted and the data string is always five characters in length including the decimal point.
For negative numbers the minus sign replaces the decimal point, eg:

+ 5.3	5.300
	05.30
	005.3
-5.3	5-300
	05-30
	005-3 etc.

Again, when returning values the 800 will always place the decimal point (or minus sign) in the same position as the instrument display and will pad out to five characters with zeros.
The choice of Free/Fixed is done by clearing/setting a "flag" or bit in a special parameter called the STATUS WORD (Mnemonic SW).

1.2.5 Status Words

These enable the computer to check or alter the various functions at the controller, eg.

Check state of alarms
Auto/Manual Selection
Remote/Local Selection
Disable front panel keys
Free or Fixed data format selection etc

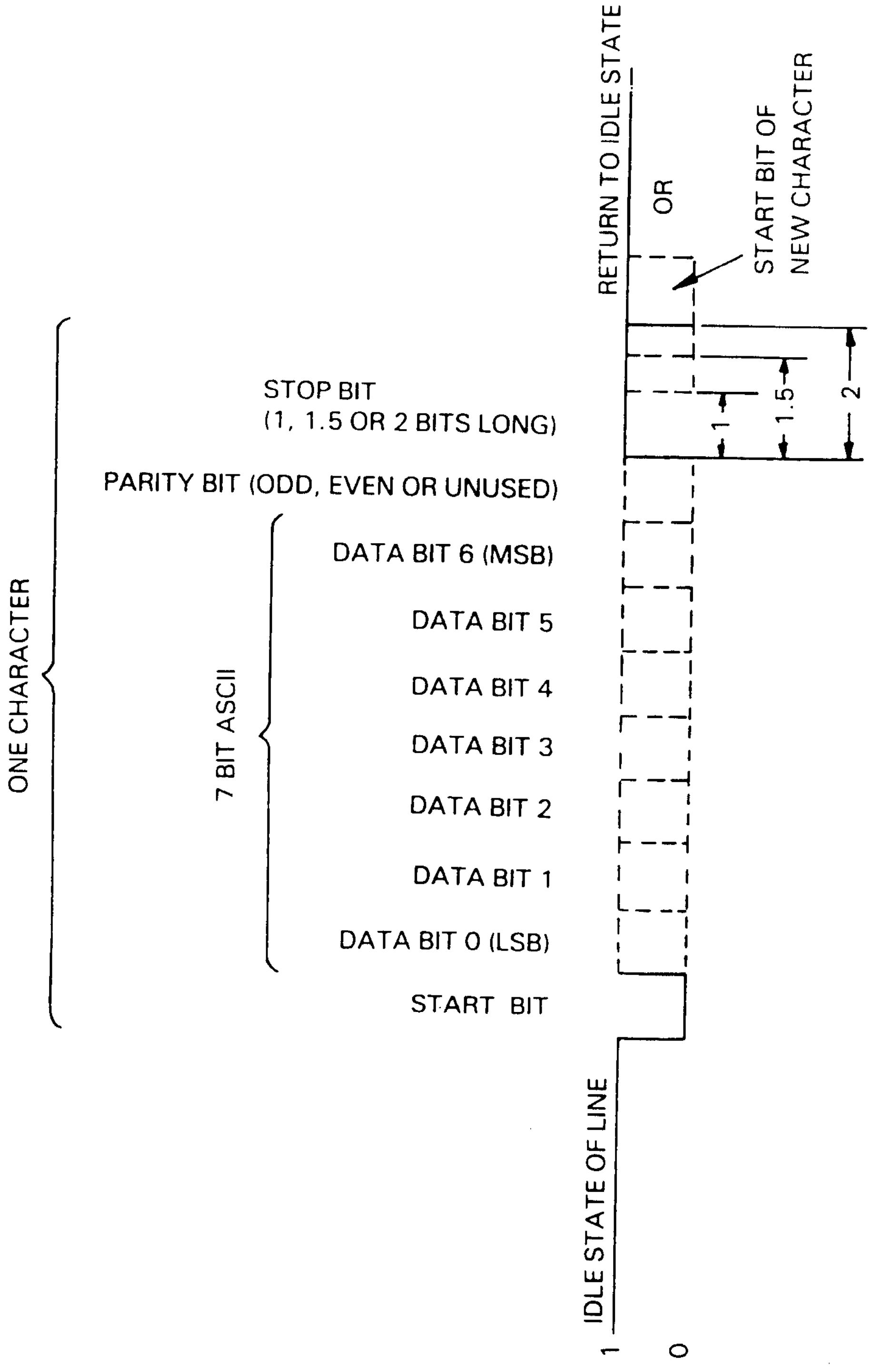
This is all detailed in Sections 4.2, 6.2, 7.2.

1.2.6 ASCII Codes

Before a character is transmitted it is turned into an ASCII code. This is universal code and a full list is included in appendix 1.
The ASCII code is 7 bits and to this the start, parity and stop bit have to be added as shown in the attached diagram. Eurotherm's protocol requires even parity and a single stop bit. This has to be programmed as part of the software concerning the communications output port of the computer.

Note: The 110 baud rate on the 820/825 instruments has two stop bits.

Asynchronous ASCII Character Format



2.0 SEQUENCE TO READ INFORMATION FROM THE 800 BY COMPUTER

2.1 Enquiry

The computer initially has master status, with the 800 in slave status and begins by transmitting a message, known as the "establish connection" message, which is represented by the following format:

(EOT) (GID) (GID) (UID) (UID) (C1) (C2) (ENQ)

These symbols are defined as follows:

(EOT) - This control character resets all instruments on the link and causes them to examine the next four transmitted characters to see if they correspond with their group/unit address identifiers.

(GID) - These characters represent the required group address identifier, repeated for security.

(UID) - These characters represent the required unit address identifier, repeated for security.
(Together these units define the address of a particular instrument).
If, for example, GID = 3 and UID = 4, then the instrument to be addressed is number 34.

(C1)(C2) - These characters specify the parameter by mnemonic.

(ENQ) - This character indicates the end of the message, and that it is an enquiry.

The transmission of this message initiates a response procedure from the 800.

2.2 Valid Response of the 800 to this Message

(For no response see 2.4)

After the message has been sent, the computer adopts slave status and expects to receive a reply from the 800. In so doing, the 800 assumes Master status and providing the 800 has successfully received the message in full, it responds in the following form:

(STX) (C1) (C2) (D1) (D2) (D3) (D4) (D5) (ETX) (BCC)

which consitutes a message defined as thus:

(STX) - start of text

(C1)(C2) - parameter specified by mnemonic

(D1 to D5) value of the requested parameter (may be up to (D6)

(ETX) - termination of text.

(BCC) - verification digit which is the character generated by taking the exclusive OR of the ASCII values of all the characters transmitted after and excluding (STX) up to and including (ETX).

ie: (BCC) = (C1) EOR (C2) EOR (D1) EOR (D2) EOR (D3) EOR (D4) EOR (D5) EOR (ETX)

where EOR = Exclusive OR

The computer must check this (BCC) before accepting this reply as valid. Also the software must be able to extract the number from the data string taking leading spaces or zeros into account.

NOTE: If the 800 receives the message but does not recognise the mnemonic it will respond with (STX) (C1) (C2) (EOT). The (EOT) hands back control to the computer.
If the 820 series instrument recognises the mnemonic but the parameter fails the internal checksum the 820 responds with (STX) (C1) (C2) ("?") (ETX) (BCC).

2.3 Further Enquiry and Termination

The computer then assumes master status again and three options are available:-

i) Repeat Parameter Facility (NAK)

If the computer transmits a (NAK) after the valid reply, it causes the 800 to repeat the parameter that was just received. This allows continuous monitoring of the same parameter without having to re-establish the connection.

ii) Scroll Mode Facility (ACK)

If the computer transmits a (ACK) after a 'valid reply', it causes the 800 to fetch the next parameter from the parameter list. This facility enables the computer to continuously sequence through all the parameters of the 800.

iii) Terminate Communication (EOT)

The termination procedure is entered when the selection of a particular instrument is no longer required or when an 800 does not respond to a message or replies with an (EOT) character. The computer assumes Master status and transmits an (EOT) character to enable all the instruments on the data link to be responsive to the next GID-UID address parameter.

Note: If the computer fails to take one of these three options, with the 820 series, the 820 will time-out after two minutes.

2.4 No Response

Under certain circumstances the computer may not receive a response from the 800. This could be due to any of the following reasons:

i) Group/Unit address identifiers not recognised

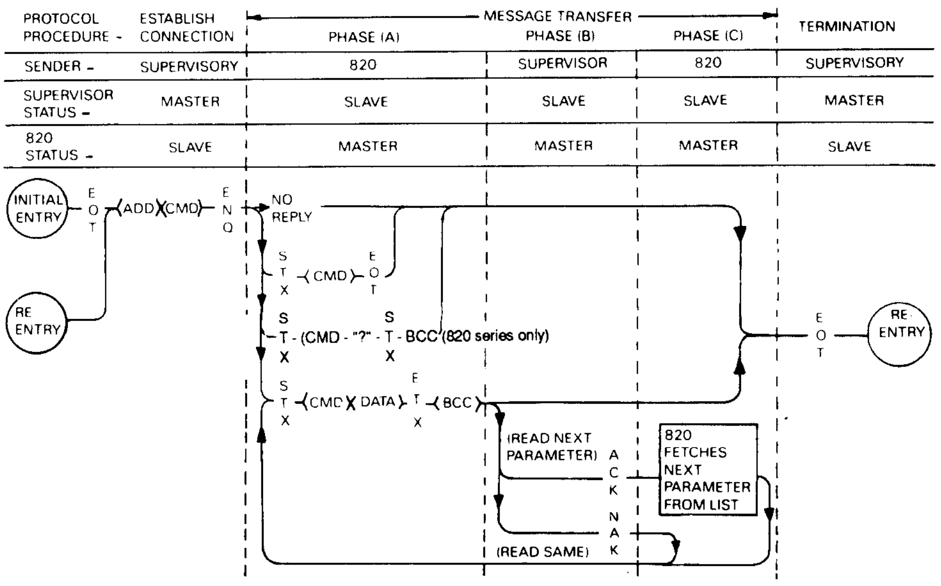
ii) An error (eg parity) is found in one or more of the characters upto and including ENQ.

iii) Communications loop failure perhaps due to noise or wrong baud rate being selected.

iv) Hardware failure.

In these cases the computer should be programmed to 'time-out' ie, wait for a response for a short time (160ms minimum) before trying again.

SELECTION SEQUENCE FOR TRANSMITTING DATA FROM THE 800 CONTROLLER TO THE SUPERVISOR.



PROTOCOL PROCEDURE -	ESTABLISH CONNECTION	MESSAGE TRANSFER PHASE (A)	PHASE (B)	PHASE (C)	TERMINATION
SENDER -	SUPERVISORY	820	SUPERVISOR	820	SUPERVISORY
SUPERVISOR STATUS -	MASTER	SLAVE	SLAVE	SLAVE	MASTER
820 STATUS -	SLAVE	MASTER	MASTER	MASTER	SLAVE

INITIAL ENTRY
EOT
(ADD)(CMD)
ENQ
NO REPLY
STX (CMD) EOT
STX - (CMD - "?" - STX - BCC (820 series only)
STX (CMD)(DATA) ETX (BCC)
(READ NEXT PARAMETER) ACK
820 FETCHES NEXT PARAMETER FROM LIST
NAK
(READ SAME)
RE ENTRY
EOT
RE ENTRY

3.0 SEQUENCE TO SEND INFORMATION TO THE 800 FROM THE COMPUTER

3.1 Establish Connection

Connection is established with a particular 800 by sending

(EOT) (GID) (GID) (UID) (UID) followed immediately by the data transfer

(STX) (C1) (C2) (D1) (D2) (D3) (D4) (D5) (D6) (ETX) (BCC)

(It will be noted that this message is identical to that transmitted by an 800 when giving a "valid reply").

The symbols of this message are defined as follows:

(STX) - start of text character
(C1)(C2) - parameter specified by mnemonic
(D1 to D5/6) - parameter value
(ETX) - end of text character
(BCC) - Block Check Character (verification check digit which is again the exclusive - OR of (C1) to (ETX) inclusive and must be calculated by the computer before transmission.

3.2 Responses

After transmission of the whole message, the 800 responds to it by sending (ACK), (NAK) or by giving no reply.

i) Positive acknowledgement (ACK)

When the 800 has received the message, it performs the following tasks:

Checks for any parity errors in the message. If none then it...
Verifies that the (BCC) character corresponds to the data pattern received. If no error then it...
Verifies that the (C1), (C2) command characters are a valid mnemonic that may be written to. If so then it....
Verifies that the data (D1 to D5) is valid and not out-of-range. If so then it...
Updates the selected parameter with the new value contained in the message.
Only when all these tasks have been successfully completed does the 800 send the (ACK) response to the computer. This signifies that the message was correctly received and implemented.

ii) Negative acknowledgement (NAK)

If the message fails any of the above checks, the 800 sends (NAK) response to the computer. This signifies that the message received by the 800 contained an error and accordingly it has not updated the selected parameter.
One possible reason is the incorrect calculation of (BCC).
At this point, the selected command may be repeated by sending the data transfer string without re-establishing connection, until the (ACK) response is received by the computer.

iii) No Reply

Under certain circumstances, the computer may not receive a response from the 800. This could be due to any of the following reasons:

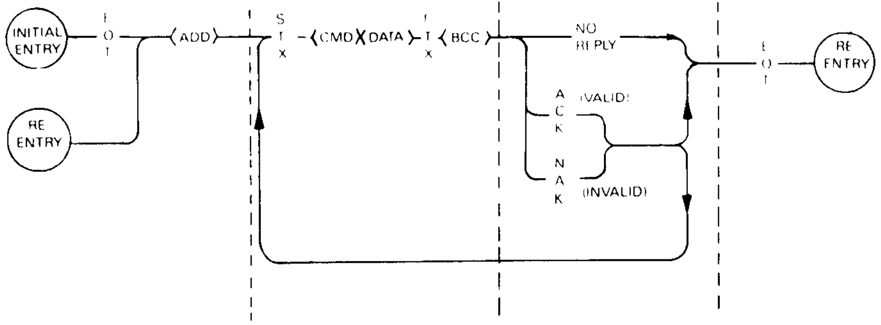
Group/Unit address identifiers not recognised.
An error (eg. parity) is found in one or more of the characters up to and including(BCC).
Communications loop failure perhaps due to noise or wrong baud rate selected.
Hardware failure.
In these cases the computer should be programmed to 'time-out', ie wait for a response for a short time (160ms minimum) before trying again.

3.3 Termination

The termination procedure is used if the computer wishes to stop selecting a particular 800 and establish connection with another. This is achieved by sending the 'establish connection' sequence. The computer retains Master status and transmits an (EOT) character to reset all instruments on the data link to be responsive to the next GID-UID address parameter.

SELECTION SEQUENCE FOR TRANSMITTING DATA FROM THE SUPERVISOR TO THE 800 CONTROLLER

		MESSAGE TRANSFER		
PROTOCOL PROCEDURE -	ESTABLISH CONNECTION	PHASE (A)	PHASE (B)	TERMINATION
SENDER -	SUPERVISORY	SUPERVISOR	820	SUPERVISOR
SUPERVISOR STATUS -	MASTER	MASTER	MASTER	MASTER
820 STATUS -	SLAVE	SLAVE	SLAVE	SLAVE



4.0 820, 821, 822 and 825 INSTRUMENTS

Baud Rate - These instruments support an extra baud rate of 110 with two stop bits.

4.1 Mnemonics

The latest up to date list of mnemonics can be found in the appropriate instrument handbook.
List of parameters available via the serial communications port of the 820/825. They are listed here in the order they would be returned if sequentially polled. This is a complete list. Some parameters are configuration specific and will not always exist within a particular instrument. The 820/825 will respond with a 'NAK' if an attempt is made to write to non-existent parameter.Versions of 820's prior to version 7 will be a subset of these mnemonics.

ASCII HEX	MNEMONIC	PARAMETER	ATTRIBUTE
50 56	PV	Process Variable 1	Read Only
53 50	SP	Active Setpoint	Read Only
45 52	ER	Error (absolute: PV-SP)	Read Only
53 56	SV	Process Variable 2	Read Only
44 52	DR	Derived Ratio	Read/Write
4F 50	OP	Output Demand (see note 1)	Read/Write
53 57	SW	Status Word (Hexadecimal)	Read/Write
4F 53	OS	Optional Status Word (Hexadecimal)	Read/Write
58 53	XS	Extended status word (Hexadecimal)	Read/Write
53 4C	SL	Internal Setpoint 1	Read/Write
4C 32	L2*	Internal Setpoint 2	Read/Write
52 49	RI	Remote Setpoint	Read/Write
52 54	RT	Remote Setpoint Trim	Read/Write
4F 41	1A	Alarm Setpoint 1	Read/Write
32 41	2A	Alarm Setpoint 2	Read/Write
48 4F	HO	Output 1 Maximum Limit	Read/Write
4C 4F	LO	Output 2 Maximum Limit	Read/Write
4F 52	OR	Output Rate Change Limit	Read/Write
48 53	HS	Setpoint Limit High	Read/Write
4C 53	LS	Setpoint Limit Low	Read/Write
48 32	H2	Setpoint 2 Limit High (Cascade)	Read/Write
4C 32	L2*	Setpoint 2 Limit Low (Cascade)	Read/Write
52 42	RB	Setpoint Bias (Ratio)	Read/Write
58 50	XP	Proportional Band 1	Read/Write
54 49	TI	Integral Time 1	Read/Write
4D 52	MR	Manual Reset 1	Read/Write
54 44	TD	Derivative Time 1	Read/Write
44 42	DB	On/Off Deadband	Read/Write
52 47	RG	Relative Cool Gain 1	Read/Write
50 32	P2	Proportional Band 2	Read/Write
49 32	I2	Integral Time 2	Read/Write
52 32	R2	Manual Reset 2	Read/Write
44 32	D2	Derivative Time 2	Read/Write
47 32	G2	Relative Cool Gain 2	Read/Write
48 42	HB	Cutback High	Read/Write
4C 42	LB	Cutback Low	Read/Write
48 43	HC	Heat/Cool Deadband	Read/Write
43 48	CH	Output 1 Cycle Time	Read/Write
43 43	CC	Output 2 Cycle Time	Read/Write
49 46	IF	Input Filter	Read/Write
42 50	BP	Default Input 1 break output	Read/Write
32 42	2B	Default Input 2 break input	Read/Write
50 45	PE	Input 1 Pyrometer Emmissivity	Read/Write
32 45	2E	Input 2 Pyrometer Emmissivity	Read/Write
53 43	SC	Security Code	Read/Write
56 30	V0	Software Version	Read/Write
49 49	II	Instrument Identifier - Hexadecimal	Read Only
31 48	1H	Display Maximum	Read Only
31 4C	1L	Display Minimum	Read Only

* Mnemonic L2 will be S2 in all future software releases.
NOTE:1. When in automatic the output power is read only. When in manual output power is read/write.
NOTE:2. SP is the working setpoint and may be SL, R I, SL + RT, L2 and it may not exceed the limits HS, LS.

4.1.1 Diagnostic Mnemonics (820 v 7.0 onwards and 825 only)

The following list describes the diagnostic parameters available via the serial communications. The event of reading the first parameter in the list forces the 820/825 to copy from the PID work area, all other parameters to the SNAPSHOT ram area. This means that an instantaneous picture of the PID working registers can be taken and read without the PID routine updating the parameters during the read sequence.

ASCII/HEX	Mnemonic	Parameter
2A 41	*A	Normalised Error (plus snapshot initiate)
2A 42	*B	Integral Output (Loop1)
2A 43	*C	Derivative Output (Loop 1)
2A 44	*D	Integral Input (Loop 2 if cascade)
2A 45	*E	Derivative Input (Loop 2 if cascade)
2A 46	*F	Control Signal (Loop 2 if cascade)
2A 47	*G	Excess Control Signal Request(Loop 2 if cascade)
2A 48	*H	Power Feedback
2A 50	*P	Normalised Error (Loop 2)
2A 51	*Q	Integral Output (Loop 2)
2A 52	*R	Derivative Output (Loop 2)
2A 5A	*Z	Internal CJC reading (Degrees C)

4.2 Status Words

These are made up of sixteen bits, each one of which represents a particular function on the controller as detailed below. These are Read /Only (R/O), Read/Write (R/W) or Read/Clear (R/C)

The 16 bits are sent as four hexadecimal numbers preceded by ">" to warn the computer that the data is hexadecimal. The format is thus SW = >ABCD. These digits are still transmitted in ASCII code.

STATUS WORD - Mnemonic SW

SW DIGIT	BIT	DESCRIPTION	ATTRIBUTE	CLEAR/SET
D	0	Data Format	R/W	Free/ Fixed
D	1	Sensor Break (I/P 1)	R/O	No/Yes
D	2	Keylock	R/W	Keys Enabled/Disabled
D	3	Checksum	R/O	OK/Failure
C	4	Setpoint Limit	R/O	In Range/Limited
C	5	Parameter changed via keys	R/C	No/Yes
C	6	Spare	-	
C	7	Spare	-	
B	8	Alarm 2 State	R/O	Off/On
B	9	Alarm 2 Cause	R/O	No Alarm 2/Alarm 2
B	10	Alarm 1 State	R/O	Off/On
B	11	Alarm 1 Cause	R/O	No Alarm 1/Alarm 1
A	12	Alarm Acknowledge	R/C	No Alarm/New Alarm 1 or 2
A	13	SP &PID Select	R/W	PID1 & SP1/PID2 & SP2
A	14	Local/Remote	R/W	Local/ Remote
A	15	Auto/Manual	R/W	Auto/Manual

Example :- SW = >8004 means the instrument is in manual, Local, PID1, No alarm, etc.....checksum OK, Keylock active, No sensor break and free data format.

Notes:
Alarm State, Alarm Cause and Alarm Acknowledge.
These are best described as follows;

Non-Latching Alarms

Alarm Condition	Absent	Present	Absent
State	0	1	0
Cause	0	1	0
Ack	0	1	0

Note: The Ack bit is set if either Alarm 1 or 2 is present.

Latched Alarms

Alarm Condition	Absent	Present	Absent
State	0	1	1
Cause	0	1	0
Ack	0	1	1*

* Clearing this will clear the State.

Optional Status Word **Mnemonic OS (Format is OS>ABCD)**

SW DIGIT	BIT	DESCRIPTION	ATTRIBUTE	CLEAR/SET
D	0	Value of D is used to control	R/W	
D	1	programmes on 821, 822	R/W	
D	2	See 4.3 below	R/W	
D	3		R/W	
C	4	Spare		
C	5	Spare		
C	6	Spare		
C	7	Spare		
B	8	Spare		
B	9	Spare		
B	10	Spare		
B	11	Spare		
A	12	Spare		
A	13	Dig Out	R/W	Off/On
A	14	Dig In 2	R/O	Off/On
A	15	Dig In 1	R/O	Off/On

4.3 821,822 Programme Selection and Control

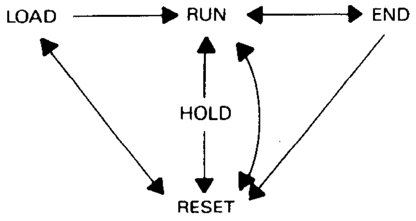
The 821 and 822 use the Optional Status Word and two extra mnemonics for program control.

The **value** of the nibble D in OS is used to control the program as follows:-

Value of D

0.	RESET	The programmer is in the reset state ie the instrument is operating as a controller.
1.	LOAD	A program has been loaded but it has not been started.
2.	RUN	A program is running.
3.	HOLD	A running program is in HOLD.
4.	END	A program has been run and it is now in the END state.

When writing to the Options Status word only some of the possible changes of state are allowed; the following diagram shows the permissible changes.



Note: 1.Changes from Reset to Load or Run are not allowed if the programme selected with the CP mnemonic is empty.
2. A running programme may be made to jump straight to END using OS>0004. The CS mnemonic can also achieve this but only by stepping through the programme segment by segment.

821/822 Program Mnemonics

There are further mnemonics which allow the 821/822 programmes to be selected and run via the link.

ASCII/HEX	MNEMONICS	FUNCTION
43 53	CS	Current Segment being executed READ/WRITE

If no program is running or the running program is in the LOAD or END state, reading with this mnemonic will return zero. It is possible to perform a 'next segment' function with the mnemonic by setting the segment to be one greater than the current segment.

43 50	CP	Current Programme READ/WRITE

The current program may only be changed when the instrument is in the RESET program state.

4.4 Instrument Identity

The mnemonic II (Read only) returns four hexadecimal characters to define the instrument type;

eg II (=ABCD) where

ABC = 820/ 825 etc
D = 0 for Production Versions of S/W
A to E for Special Versions
F for Unreleased Prototype Software

Software Version (V0)

This will consist of four hex characters in the format >ABCD

A = Software Class
0 to 9 Production Release
A to E Specials
F Prototype

B = Software Version
0 to 9
CD = Software Issue
00 to 99

4.5 Datalog Option

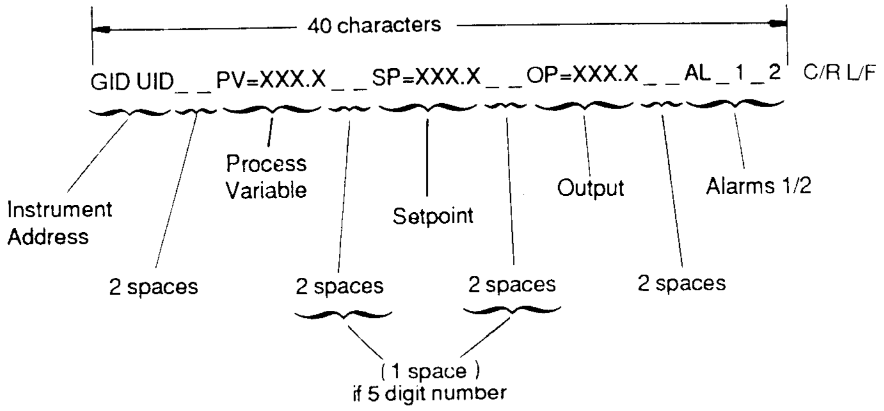
The 820 Data Log facility is selected via the configuration mode. The baud rate of the communications link is also selected during configuration, as is the instrument address (group member). The instrument prefixes all Data Log messages with its address for identification.

The rate at which the instrument dumps the Data Log message can be controlled automatically, by setting the LOG RATE . This is done in the commissioning mode. The LOG RATE can be set at 1 minute intervals between 1 minute and 20 minutes per sample. A LOG RATE of 0 disables the Data Log function.

The Data Log facility is.enabled by either linking the Enable negative input to common, or taking the Enable positive input, terminal 14, above 2V (ref. digital common). As long as the described conditions persist, the instrument will continue to Data Log at the selected rate. The first Data Log message is initiated by the action of applying the Data Log enable signal.

The Data Log message can also be controlled externally simply by using the Data Log enable signal to control the LOG RATE. The enable signal can be applied to either ENABLE +, terminal 14, or ENABLE -, terminal 15, to generate a maximum LOG RATE, per logged message, of 5 seconds (110 baud) or 0.25 seconds (9600 baud). The enable signal must be active for more than 20mS, for the signal to be detected.

Example of Datalog format:



5.0 822 PROGRAMME COMMUNICATIONS PROTOCOL.

Note 1: For programme upload or download, the communications microprocessor software must be Version 7 or later, main software must be version 5.1 or later.
Note 2: Selecting and running programmes is controlled by via the optional Status Word section 4.3.
Note 3: It is not possible to communicate programme variables with the 821.

5.1 GENERAL.

The format of the protocol used for changing and reading programme variables is different from the normal parameter and status word communications as it involves accessing the RAM of the 822 by using relative addresses starting from 000, which is the start point of a specific programme memory as allocated internally by the 822. As a result the host computer has to be able interpret the blocks of information and data returned during a READ sequence and, more significantly, has to be able to create new program data into the correct BLOCK format, including the allocation of the correct relative RAM addresses, when performing a programme WRITE sequence. Before a WRITE sequence it is necessary to check that there is sufficient RAM space available otherwise the program will not be accepted.

The format and sequence of blocks of information is the same for READ and WRITE. If the host READS a programme already in the 822 and stores the returned information in exactly the same format, it may be later written back to the 822, in a different location, as any selected programme number.

The 822 must be in RESET (controller only mode) and the operator must not be in the programme entry mode via the front panel for the READ and WRITE to work via the communications link. If the link fails during a READ or WRITE sequence, the 822 times out 4 seconds after the last valid message. The complete sequence will then have to be restarted from the beginning.

5.2 MNEMONICS.

Five new mnemonics are used to permit programmes to be transfered to and from a host computer:

ASCII-HEX	MNEMONIC		ATTRIBUTE
4D 46	MF	Poll number of memory locations free	READ only
42 44	BD	Begin download (WRITE) sequence	WRITE only
45 4E	EN	End programme write sequence	WRITE only
42 55	BU	Begin programme upload (READ) sequence	WRITE only
4B 50	KP	Kill (delete) a programme	WRITE only

Note: BD mnemonic deletes the selected programme if it already exists.

5.3 BLOCK FORMAT.

The various BLOCKS are all defined in section 5.5.

The following BLOCK format is used to allow the transfer of information:

@AAADDDDDDD where

@ signifies a programme READ/WRITE sequence

AAA is a three digit hexadecimal relative address of where the programme resides in RAM which always starts from 000.

DDDD is either between 1 to 7 hexadecimal characters of programme information or

a free format decimal number of programme data (the same as parameter values).

The BLOCK is always preceded by (STX) and is concluded with (ETX) and the usual (BCC) - the Exclusive OR of all characters after and excluding (STX) up to and including (ETX).

5.4 BLOCK SEQUENCE STRUCTURE

This core is the same for READ and WRITE sequences. There are 11 types of BLOCKS and these are all defined in Section 5.5. Some types of BLOCK are mandatory and some will depend on the programme features. The number of memory locations taken is also listed.

Type Of Block	**Function**	**Data Hex or Decimal**	**Mandatory (M) Prog. depend (Pd) Segm. depend (Sd)**	**Memory Location Hex**
1	**PROGRAM HEADER**	Hex(7)	M	7
2	Extension header	Dec (Level	Pd	6
3	Extension header	Dec (End level)	Pd	6
4	Extension header	Dec (Loop Count)	Pd	4
5	**SEGMENT 1 HEADER**	Hex(1)	M	1
6	Segment options	Hex(2)	Pd	2
7	Segment options	Hex(1)	Pd	1
8a	Segment options	Dec (special P)	Pd	6
8b	Segment options	Dec (special I)	Pd	6
8c	Segment options	Dec (special D)	Pd	6
9	Segment options	Dec (Holdback value)	Pd	6
10	**Parameter Value**	Dec	M	6
		Hex(1) for subprog		1
11	2nd Par. value	Dec (Ramp Level)	Sd	6
5	**SEGMENT 2 HEADER**	Hex(1)	Pd	
to				
11	as required (minimum per segment is BLOCKS 5 and 10)			
5	**SEGMENT 3 HEADER**		Pd	
etc	(up to 16 segments)		Pd	

5.5 BLOCK DESCRIPTIONS.

Each TYPE of BLOCK takes a certain number of memory locations and this number is indicated against each BLOCK description.

5.5.1 BLOCK TYPE 1: PROGRAMME HEADER BLOCK

Hex Locations required 7

@AAA

Start of relative address in RAM in HEX. Must be 000.

DDD

Size, in HEX, of memory required for the programme. Sum of the number of locations required by all the BLOCKS, including this header, necessary to define the programme.

eg send "1A4" for programme requiring 1A4 (HEX) or 420 (decimal) locations.

D

Programme number HEX 0= programme 1, F= programme 16.

eg send "3" for programme 4.

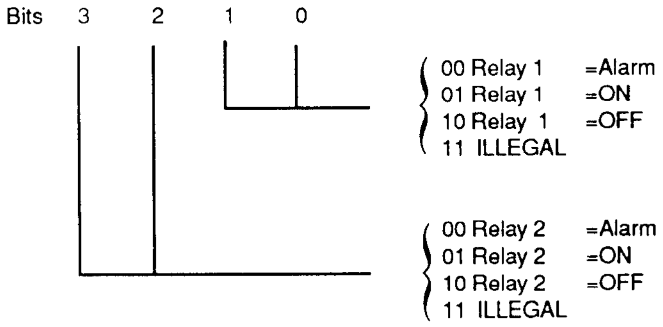
D

Total number of segments HEX 0=1 segment, F= 16 segments

eg send "A" for 11 segment programme

D

Relay Options in the END state. HEX used as bits 3, 2, 1, 0.



Bits 3 2 1 0

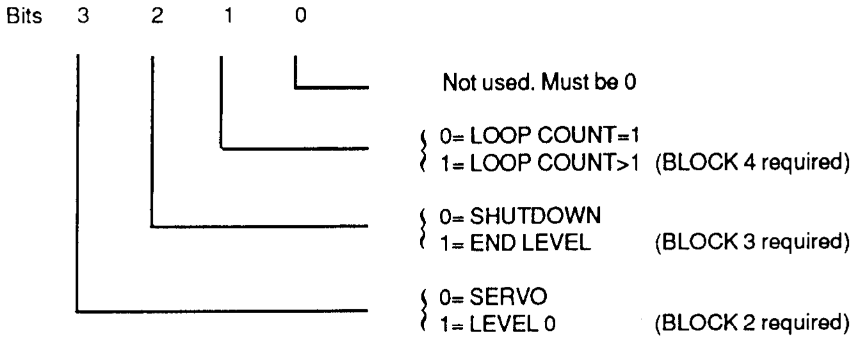
00 Relay 1 =Alarm
01 Relay 1 =ON
10 Relay 1 =OFF
11 ILLEGAL

00 Relay 2 =Alarm
01 Relay 2 =ON
10 Relay 2 =OFF
11 ILLEGAL

eg send "8" (1000) for Relay 1 as alarm, Relay 2 OFF during the END state.

D

START / END options.HEX as bits 3, 2, 1. Bit 0 must be 0.



Bits 3 2 1 0

Not used. Must be 0

0= LOOP COUNT=1
1= LOOP COUNT>1 (BLOCK 4 required)

0= SHUTDOWN
1= END LEVEL (BLOCK 3 required)

0= SERVO
1= LEVEL 0 (BLOCK 2 required)

eg send "0" for SERVO, SHUTDOWN and a LOOP count of 1, and the complete block in examples shown is @ 000 1A43A8.

5.5.2 BLOCK TYPE 2: LEVEL 0 **if selected in BLOCK 1**

Hex Locations required 6

@AAA

Relative address calculated by taking the address from the previous BLOCK and adding to it the number of locations that BLOCK required.

Followed by free format decimal data. Value must lie between Setpoint High and Low.

5.5.3 BLOCK TYPE 3: END LEVEL **if selected in BLOCK 1**

Hex Locations required 6

@AAA

Relative address, as before.

Followed by free format decimal data. Value must lie between Setpoint High and Low.

5.5.4 BLOCK TYPE 4: LOOP COUNT if >1 is selected in BLOCK 1

Hex Locations required 4

@AAA Relative address as before.

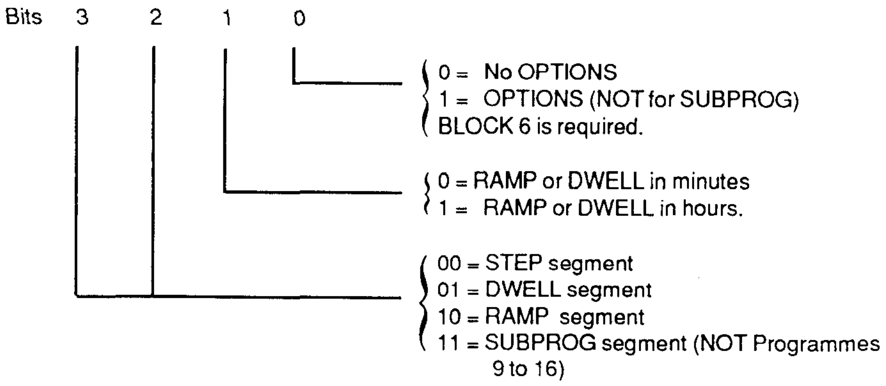
Followed by free format data. Value must lie between 2 and 9999.

5.5.5 BLOCK TYPE 5: SEGMENT HEADER

Hex Locations required 1

@AAA Relative address, as before.

D Single HEX used as Bits 3, 2, 1, 0 to define the segment type:



Bits 3 2 1 0

0 = No OPTIONS
1 = OPTIONS (NOT for SUBPROG)
BLOCK 6 is required.

0 = RAMP or DWELL in minutes
1 = RAMP or DWELL in hours.

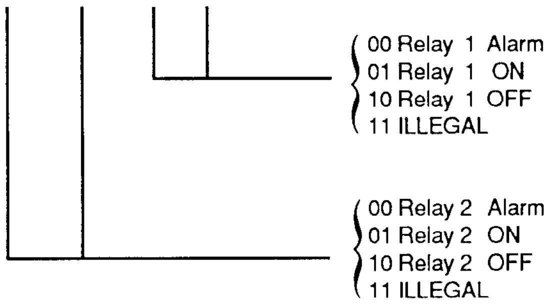
00 = STEP segment
01 = DWELL segment
10 = RAMP segment
11 = SUBPROG segment (NOT Programmes 9 to 16)

eg send "5" (0101) for a DWELL in minutes with options.
send "A" (1010) for a RAMP in units/hour without options.
send "C" (1100) for a subprogram call segment.

5.5.6 BLOCK TYPE 6 : SEGMENT OPTIONS If selected In BLOCK 5. Hex Locations required 2

@AAA Relative address, as before.

D Relay Options in the segment. HEX used as bits 3, 2, 1, 0.
Bits 3 2 1 0



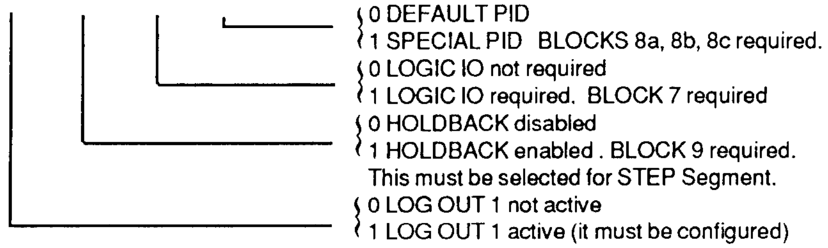
00 Relay 1 Alarm
01 Relay 1 ON
10 Relay 1 OFF
11 ILLEGAL

00 Relay 2 Alarm
01 Relay 2 ON
10 Relay 2 OFF
11 ILLEGAL

eg send "4" (0100) for Relay as Alarm, Relay 2 ON during the segment.

D Selects other OPTIONS during a segment. HEX used as Bits 3, 2, 1, 0.

Bits 3 2 1 0



0 DEFAULT PID
1 SPECIAL PID BLOCKS 8a, 8b, 8c required.
0 LOGIC IO not required
1 LOGIC IO required. BLOCK 7 required
0 HOLDBACK disabled
1 HOLDBACK enabled . BLOCK 9 required.
This must be selected for STEP Segment.
0 LOG OUT 1 not active
1 LOG OUT 1 active (it must be configured)

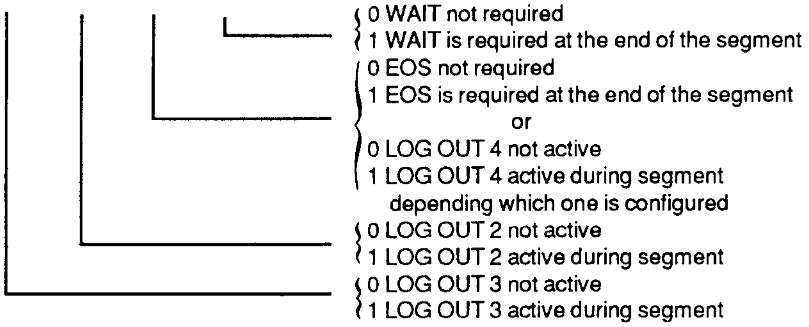
e g send " 5" (0101) for segment with Holdback and special PID, but no LOGIC IO functions.

5.5.7 BLOCK TYPE 7: LOGIC IO if selected in BLOCK 5. Hex Locations required 1

@AAA Relative address as before

D Selects the various Logic outputs provided they are configured in the instrument. HEX used as Bits 3, 2, 1, 0.

Bits 3 2 1 0



0 WAIT not required
1 WAIT is required at the end of the segment
0 EOS not required
1 EOS is required at the end of the segment
or
0 LOG OUT 4 not active
1 LOG OUT 4 active during segment
depending which one is configured
0 LOG OUT 2 not active
1 LOG OUT 2 active during segment
0 LOG OUT 3 not active
1 LOG OUT 3 active during segment

e g send "C" (1100) for LOG OUT 2 and 3 active during segment.

5.5.8 BLOCKS TYPES 8a, 8b, 8c:SPECIAL PID if selected in BLOCK 5.

All three BLOCKS must be sent.

Hex Locations required 3 x 6 = 12 Hex

The exact function of each BLOCK depends on instrument configuration;

Instrument Configurations	8a	8b	8c
on/off	Deadband	0.0	0.0
PD	Xp	Manual Reset	Td (Ti MUST = 0.0)
PID	Xp	Ti	Td

BLOCK TYPE 8a. @AAA Relative address, as before. Followed by free format decimal data, the value within the appropriate range.

BLOCK TYPE 8b @AAA Relative address, as before. Followed by free format decimal data, the value within the appropriate range.

BLOCK TYPE 8c @AAA Relative address, as before followed by free format decimal data, the value within the appropriate range.

5.5.9 BLOCK TYPE 9; HOLDBACK if selected in BLOCK 5.

Hex Locations required 6.

@AAA Relative address. Followed by the free format decimal data. The value must lie within the instrument span (display max - display min).

5.5.10 BLOCK TYPE 10: SEGMENT PARAMETER VALUE.

This must be sent for each segment.

Hex Locations required SUBPROG (HEX) 1
All other types 6

@AAA Relative address followed by the free format decimal value of the RAMP rate, DWELL time, or STEP to value within the appropriate range.

For a SUBPROG segment a single HEX character is used in the range 8 to F, where 8 is programme 9 and F programme 16.

5.5.11 BLOCK TYPE 11: RAMP TO VALUE is required for RAMP segments.

Hex Locations required 6

@AAA Relative address followed by free format decimal value of the target to which the setpoint is ramping between setpoint high and setpoint low.

5.6 PROGRAMME READ SEQUENCE

Computer: Sends instrument address, 'BU' mnemonic followed by number of the programme to be read. (In the usual parameter write format. See section 3.1).

822 Responds with (ACK) if it is in reset and the programme requested exists and programmes are not being edited manually, otherwise (NAK).

Computer: If it receives an (ACK), it disengages using (EOT), then within 4 seconds sends instrument address, @000, followed by (ENQ). (Similar to the usual parameter read format).

822: Sends BLOCK 1
Computer: (ACK)

822: Sends next BLOCK in the sequence described above.

etc until

822: Sends final BLOCK
Computer: (ACK)

822: Sends a BLOCK with no data (i.e. (STX) @ AAA (ETX) (BCC) only).

Computer: (EOT) to terminate the transaction as it recognizes the lack of data in the last BLOCK. The computer can terminate the read at any time by responding not with (ACK) but by writing the "EN" mnemonic followed by the programme number.

5.7 SIMPLE PROGRAMME READ EXAMPLE

To READ programme 2 from an 822 with address 0.0

Computer: (EOT) 0 0 0 0 (STX) B U 2. (ETX) (BCC) : Initiate prog 2 upload
822: (ACK)

Computer: (EOT) 0 0 0 0 @ 0 0 0 (ENQ)
822: (STX) @ 0 0 0 0 2 8 1 2 0 0 (ETX) (BCC) : Block 1
Comment: Programme 2 takes 028 HEX locations
has 3 segments
starts with SERVO
ends with SHUTDOWN
has LOOP COUNT = 1
has both Relays as alarms in the END state.
It can be seen that BLOCKS 2, 3, 4 are not required. The next BLOCK will be 5.
Computer: (ACK)

822: (STX) @ 0 0 7 8 (ETX) (BCC) : Block 5

Comment: Note the relative address, BLOCK 1 takes 7 locations. This is the segment header for segment 1, a RAMP, in minutes and without options so BLOCKS 6, 7, 8a, 8b, 8c, 9 are not required.
Computer: (ACK)

822: (STX) @ 0 0 8 2 . 0 (ETX) (BCC) : Block 10
Comment: Segment parameter value, in this case 2.0 units per minute.
As it is a RAMP, the next BLOCK will be BLOCK 11.
Computer: (ACK)

822: @ 0 0 E 1 0 . 0 (ETX) (BCC) : Block 11
Comment: Setpoint will ramp to 10.0 units.
This is now the end of segment 1.
Computer: (ACK)

822: (STX) @ 0 1 4 4 (ETX) (BCC) : Block 5
Comment: Dwell, in minutes without options.
Computer: (ACK)

822: (STX) @ 0 1 5 1 . 0 (EXT) (BCC) : Block 10
Comment: Dwell 1.0 minute.
Computer: (ACK)

822: (STX) @ 0 1 B 8 (ETX) (BCC) : Block 5
Comment: Segment 3 is a RAMP in minutes without options.
Computer: (ACK)

822: (STX) @ 0 1 C 1 . 0 (ETX) (BCC) : Block 10
Comment: Ramp at 1.0 units per second.
Computer: (ACK)

822: (STX) @ 0 2 2 2 0 . 0 (ETX) (BCC) : Block 11
Comment: Ramp to 20.0 units.
Computer: (ACK)

822: (STX) @ 0 2 8 (ETX) (BCC) : End (No data)

Comment: There is no data so the Upload is complete. Note that the final relative address, 028 hex, is the value of the total number of locations required as was included in BLOCK 1. The 822 has no more data to send so the computer should now end the sequence:

Computer: (EOT).

5.8 PROGRAMME WRITE SEQUENCE

Before starting the write sequence the host computer should poll the 822 using the "MC" mnemonic to ensure that there is enough free RAM space for the programme. The memory space for the programme should be established by the addition of the memory locations required for all the BLOCKS necessary to define the programme.

Computer: Sends instrument address, "BD" mnemonic followed by number of the programme to be written. (In the usual parameter write format).

822: Responds with (ACK) if it is in reset and programmes are not being edited manually, otherwise (NAK).

Computer: If it receives (ACK), it disengages using (EOT) and within 4 seconds sends the instrument address followed by BLOCK 1.

Note: The programme will be stored as the programme number specified by the "BD" mnemonic above. The programme number contained in BLOCK 1 is ignored. This enables an UPloaded programme to be DOWN loaded as any of the 16 programmes without any change to the data.

822: (ACK)

Computer: = Sends next BLOCK in the sequence described above.

822: (ACK) provided all the addresses and data contain no errors such as setpoints out of range etc. in which case a (NAK) is returned. The whole sequence must then be restarted (after correction!).

etc until the computer has sent the last BLOCK.

822: (ACK)

Computer: Sends "EN" mnemonic followed by the programme number in the usual parameter write format.
822: (ACK). The 822 will now have transferred the programme into the RAM storage area.

Computer: (EOT) to terminate the connection.

The computer can terminate the write at any time by responding not with the next BLOCK but by writing the "EN" mnemonic followed by the programme number.

5.9 SIMPLE PROGRAMME WRITE EXAMPLE.

Ramp from 25 to 150 at 5.0 per hour with relay 2 off and Holdback limit of 1.0 units. Relay 1 always remains an alarm.

Dwell for 1.6 hours with relay 2 on.

End level 150 with relay 2 off.

DOWNload to programme 12 on an 822 with address 0.0.
Sequence will be:

			Relative address to be used below
a. BLOCK TYPE 1	Header	7	000
b BLOCK TYPE 2	Level 0	6	007
c. BLOCK TYPE 3	End level	6	00D
d. BLOCK TYPE 5	1st Segment header	1	013
e. BLOCK TYPE 6	Relay options	2	014
f. BLOCK TYPE 9	Holdback value	6	016
g. BLOCK TYPE10	Ramp rate	6	01C
h. BLOCK TYPE11	Target	6	022
i. BLOCK TYPE 5	2nd Segement header	1	028
j. BLOCK TYPE 6	Relay options	2	029
k. BLOCK TYPE 10	Dwell time	6	02B
		49	031

Programme takes 49 decimal or 31 HEX locations. The computer should first check that this number of locations are free.
So the download sequence looks like this:

```
     Computer:  (EOT) 0 0 0 0 (STX) B D  1 2 . (ETX) (BBC)
     822:       (ACK)      The programme will be No. 12.

a.   Computer: (EOT) 0 0 0 0 (STX) @ 0 0 0 0 3 1 B 1 8 C (ETX) (BCC)      : Block 1
     822:       (ACK)      031 is memory requirement.

b.   Computer:  (STX) @ 0 0 7  2 5 . (ETX) (BCC)                          : Block 2
     822        (ACK) Level 0

c.   Computer:  (STX) @ 0 0 D  1 5 0 . (ETX) (BCC)                        : Block 3
     822:       (ACK)      End level

d.   Computer:  (STX) @ 0 1 3 B (ETX) (BCC)                               : Block 5
     822:       (ACK) Ramp, per hour with Holdback and relay options.

e.   Computer:  (STX) @ 0 1 4 8 4 (ETX) (BBC)                             : Block 6
     822:       (ACK)    Relays

f.   Computer:  (STX) @ 0 1 6  1 . (ETX)  (BCC)                           : Block 9
     822:       ACK)     Holdback limit

g.   Computer:  (STX) @ 0 1 C  5 . 0 (ETX) (BCC)                          : Block 10
     822:       (ACK)    Ramp rate

h.   Computer:  (STX) @ 0 2 2  1 5 0 . (ETX) (BCC)                        : Block 11
     822        (ACK)    Ramp to

i.   Computer:  (STX) @ 0 2 8 7 (ETX) (BCC)                               : Block 5
     822:       (ACK)    Dwell hours with relay options.

j.   Computer:  (STX) @ 0 2 9 4 0 (ETX) (BCC)                             : Block 6
     822:       (ACK)    Relays options.

k.   Computer:  (STX) @ 0 2 B  1 . 6 (ETX) (BCC)  Dwell time.             : Block 10
     Comment: This is the last BLOCK. Note that 02 B+ 6 = 031, the original
                calculated memory size, in Hex
     822:       (ACK)

     Computer:  (STX) E N  1 2 . (ETX) (BCC)                              : Terminate Download
     822:       (ACK)  Only now is the programme accepted into the 822 RAM.

     Computer:  (EOT)
```

6.0 815 and 818 INSTRUMENTS

6.1 Parameter Mnemonics

The latest up to date list of mnemonics can be found in the appropriate instrument handbook.
The table below indicates the full list of communications mnemonics that will be supported by the production software. The order of the table is that which would be obtained if a fast poll was performed commencing from measured value (PV). The instrument will return a NAK if an attempt is made to access a non-existant parameter.

ASCII/HEX	MNEMONIC	PARAMETER	ATTRIBUTE
50 56	PV	Measured Value	Read/Only
53 50	SP	Working Setpoint	Read/Only
4F 50	OP	Output	Read/Write (R/O In AUTO)
53 57	SW	Status Word	Read/Write
4F 53	OS	'Optional' Status Word	Read/Write
58 53	XS	Extended Status Word	Read/Write
31 41	1A	Alarm 1	Read/Write
32 41	2A	Alarm 2	Read/Write
45 52	ER	Error	Read/Only
53 4C	SL	Local setpoint (setpoint 1)	Read/Write
53 32	S2	Setpoint 2Read/Write	Read/Write
4F 31	O1	Digital Output Status Word 1	Read/Write
4F 32	O2	Digital Output Status Word 2	Read/Write
4F 33	O3	Digital Output Status Word 3	Read/Write
4F 34	O4	Digital Output Status Word 4	Read/Write
54 4D	TM	Time remaining in current programme segment	Read/Only
4C 52	LR	Loops remaining for current program	Read/Only
72 31	r1	Ramp rate 1	Read/Write
6C 31	l1	Ramp level 1	Read/Write
74 31	t1	Dwell time 1	Read/Write
72 32	r2	Ramp rate 2	Read/Write
6C 32	l2	Ramp level 2	Read/Write
74 32	t2	Dwell time 2	Read/Write
72 33	r3	Ramp rate 3	Read/Write
6C 33	l3	Ramp level 3	Read/Write
74 33	t3	Dwell time 3	Read/Write
72 34	r4	Ramp rate 4	Read/Write
6C 34	l4	Ramp level 4	Read/Write
74 34	t4	Dwell time 4	Read/Write
72 35	r5	Ramp rate 5	Read/Write
6C 35	l5	Ramp level 5	Read/Write
74 35	t5	Dwell time 5	Read/Write
72 36	r6	Ramp rate 6	Read/Write
6C 36	l6	Ramp level 6	Read/Write
74 36	t6	Dwell time 6	Read/Write
72 37	r7	Ramp rate 7	Read/Write
6C 37	l7	Ramp level 7	Read/Write
74 37	t7	Dwell time 7	Read/Write
72 38	r8	Ramp rate 8	Read/Write
6C 38	l8	Ramp level 8	Read/Write
74 38	t8	Dwell time 8	Read/Write
48 62	Hb	Holdback value	Read/Write
4C 63	Lc	Loop count	Read/Write
52 52	RR	Ramp rate	Read/Write
52 49	RI	Remote input	Read/Only
48 4F	HO	Max. Heat	Read/Write
4C 4F	LO	Max. Cool	Read/Write
52 48	RH	Remote Heat Limit	Read/Only
52 43	RC	Remote Cool Limit	Read/Only
48 53	HS	Setpoint 1 maximum	Read/Only
4C 53	LS	Setpoint 1 minimum	Read/Only
32 48	H2	Setpoint 2 maximum	Read/Only
32 4C	L2	Setpoint 2 minimum	Read/Only

Cont.

ASCII/HEX	MNEMONIC	PARAMETER	ATTRIBUTE
43 4B	CH	Cycle time for channel 1	Read/Write
58 50	XP	Proportional Band	Read/Write
54 49	TI	Integral Time	Read/Write
4D 52	MR	Manual Reset	Read/Write
54 44	TD	Derivative Time	Read/Write
48 42	HB	Cutback High	Read/Write
4C 42	LB	Cutback Low	Read/Write
52 47	RG	Relative Cool Gain	Read/Write
48 43	HC	Heat Cool Deadband	Read/Write
43 43	CC	Cool Cycle Time	Read/Write
48 32	C2	Channel 2 Cycle Time	Read/Write
50 45	PE	Pyrometer Emissivity	Read/Write
42 50	BP	Power level at sensor break	Read/Write
56 30	V0	Software Version	Read/Only
49 49	II	Instrument Identity	Read/Only
31 48	1H	Display maximum*	Read/Only
31 4C	1L	Display minimum*	Read/Only

*Note : the true limits of PV are 1H + 10% of (1H-1L) to 1L - 10% of (H - 1L)

6.2 Status Words

These are made up of sixteen bits, each one of which represents a particular function on the controller as detailed below. These are Read Only (R/O), Read/Write (R/W) or Read/Clear (R/C).

The 16 bits are sent as four hexadecimal numbers preceded by ">" to warn the computer that the data is hexadecimal. The format is thus SW = >ABCD. These digits are still transmitted in ASCII code.

Digits ABCD are ASCII characters representing a hexadecimal digit (0-9, A-F).

Status Word (SW) in the format (>ABCD)

DIGIT	BIT	FUNCTION	ATTRIBUTE	CLEAR/SET
D	0	Data Format	R/W	Free/Fixed
D	1	Sensor Break	R/O	No/Yes
D	2	Key Lock	R/W	Keys Enabled/Disabled
D	3	N/A	-	
C	4	N/A	-	
C	5	Parameter changed via keys	R/C	No/Yes
C	6	N/A	-	
C	7	N/A	-	
B	8	Alarm 2 State	R/O	Off/On
B	9	N/A	-	
B	10	Alarm 1 State	R/O	Off/On
B	11	N/A	-	
A	12	Alarm Active	R/O	No Alarm/New Alarm 1 or 2
A	13	SP 2 Active	R/W	SP1/SP2
A	14	Remote Active	R/W	Local/Remote
A	15	Auto/Manual	R/W	Auto/Man

Optional Status Word (OS) in the format (>ABCD)

DIGIT	BIT	FUNCTION	ATTRIBUTES	CLEAR/SET
D	0	Prog Status	Used to control programmes see 6.3	
D	1	Prog Status		
D	2	Prog Status		
D	3	Prog Status (MSB)		
C	4	Hold Logged	R/C	Continue/Hold
C	5	Skip Current Segment	R/W	Remain/Skip
C	6	Ramp/Dwell	R/O	Ramp/Dwell
C	7	Digital Input Lock	R/W	Enable/Disable
B	8	Segment No. (LSB)	*	
B	9	Segment No.	*	
B	10	Segment No.	*	
B	11	Segment No. (MSB)	*	
A	12	Digital O/P2 (channel 4)	R/O	Off/On
A	13	Digital O/P1 (channel 3)	R/O	Off/On
A	14	Digital Input 2	R/O	Off/On
A	15	Digital Input 1	R/O	Off/On

* Segment No. is a nibble having the value 1 to 8 Read only.

Extension Status Word (XS) in the format (>ABCD)

DIGIT	BIT	FUNCTION	ATTRIBUTES	CLEAR/SET
D	0	Self Tune	R/W	Off/On
D	1	Adaptive Tune	R/W	Off/On
D		Spare		
D		Spare		
C		Spare		
C		Spare		
C		Spare		
C		Spare		
B		Spare		
B		Spare		
B		Spare		
B		Spare		
A		Spare		
A		Spare		
A		Spare		
A		Spare		

818P Communications Status Words

Output channels 3/4 (usually used for alarms) can be configured to be driven by the programmer.Each segment can then be programmed to activate the output when the program is in that particular segment. The programming of this can be achieved via digital communications using O1, O2 , O3 or O4 as detailed below.

Digital Output Status Word 1 (O1) in the format (>ABCD

DIGIT	BIT	FUNCTION	ATTRIBUTES	CLEAR/SET
D	0	ramp 1 to channel 3	R/W	activate/no effect
D	1	dwell 1 to channel 3	R/W	activate/no effect
D	2	ramp 2 to channel 3	R/W	activate/no effect
D	3	dwell 2 to channel 3	R/W	activate/no effect
C	4	ramp 3 to channel 3	R/W	activate/no effect
C	5	dwell 3 to channel 3	R/W	activate/no effect
C	6	ramp 4 to channel 3	R/W	activate/no effect
C	7	dwell 4 to channel 3	R/W	activate/no effect
B	8	ramp 5 to channel 3	R/W	activate/no effect
B	9	dwell 5 to channel 3	R/W	activate/no effect
B	10	ramp 6 to channel 3	R/W	activate/no effect
B	11	dwell 6 to channel 3	R/W	activate/no effect
A	12	ramp 7 to channel 3	R/W	activate/no effect
A	13	dwell 7 to channel 3	R/W	activate/no effect
A	14	ramp 8 to channel 3	R/W	activate/no effect
A	15	dwell 8 to channel 3	R/W	activate/no effect

Digital Output Status Word 2 (O2) in the format (>ABCD)

DIGIT	BIT	FUNCTION	ATTRIBUTES	CLEAR/SET
D	0	END to channel 3	R/W	activate/no effect
D	1	Spare		
D	2	Spare		
D	3	Spare		
C	4	Spare		
C	5	Spare		
C	6	Spare		
C	7	Spare		
B	8	Spare		
B	9	Spare		
B	10	Spare		
B	11	Spare		
A	12	Spare		
A	13	Spare		
A	14	Spare		
A	15	Spare		

Digital Output Status Word 3 (O3) in the format (>ABCD)

DIGIT	BIT	FUNCTION	ATTRIBUTES	CLEAR/SET
D	0	ramp 1 to channel 4	R/W	activate/no effect
D	1	dwell 1 to channel 4	R/W	activate/no effect
D	2	ramp 2 to channel 4	R/W	activate/no effect
D	3	dwell 2 to channel 4	R/W	activate/no effect
C	4	ramp 3 to channel 4	R/W	activate/no effect
C	5	dwell 3 to channel 4	R/W	activate/no effect
C	6	ramp 4 to channel 4	R/W	activate/no effect
C	7	dwell 4 to channel 4	R/W	activate/no effect
B	8	ramp 5 to channel 4	R/W	activate/no effect
B	9	dwell 5 to channel 4	R/W	activate/no effect
B	10	ramp 6 to channel 4	R/W	activate/no effect
B	11	dwell 6 to channel 4	R/W	activate/no effect
A	12	ramp 7 to channel 4	R/W	activate/no effect
A	13	dwell 7 to channel 4	R/W	activate/no effect
A	14	ramp 8 to channel 4	R/W	activate/no effect
A	15	dwell 8 to channel 4	R/W	activate/no effect

Digital Output Status Word 4 (O4) in the format (>ABCD)

DIGIT	BIT	FUNCTION	ATTRIBUTES	CLEAR/SET
D	0	END to channel 4	R/W	activate/no effect
D	1	Spare		
D	2	Spare		
D	3	Spare		
C	4	Spare		
C	5	Spare		
C	6	Spare		
C	7	Spare		
B	8	Spare		
B	9	Spare		
B	10	Spare		
B	11	Spare		
A	12	Spare		
A	13	Spare		
A	14	Spare		
A	15	Spare		

6.3 Programme Control on 818P

Programme control is set by the value of D in the Optional Status Word OS in section 6.2. It is also used to control the ramp on an 818S

Program Status is a nibble having the value 0 to 5:

D = 0	Reset Programmer/Ramp Function.	Read/Write
D = 1	N/A	
D = 2	Run Programmer/Ramp Function.	Read/Write
D = 3	Hold Programmer.	Read/Write
D = 4	End Programmer.	Read only
D = 5	RMP Engaged.	Read only
D = 6	Program in Holdback	Read only

Note: in particular bit 5 of OS which may be set to make the programme skip the current segment.

Examples:
Writing

OS = >0000 will reset programme.
OS = >0002 will run programme
OS = >0003 will hold programme

6.4 815, 818 and 818P Instrument Identity

Instrument Identity (II)

This will consist of four hex characters in the format >ABCD

ABC = Instrument Type 815, 818, etc.

D = Software Class
0 to 9 Production Release
A to E Specials
F Prototype

Software Version (V0)

This will consist of four hex characters in the format >ABCD

A = Software Class
0 to 9 Production Release
A to E Specials
F Prototype

B = Software Version
0 to 9

CD = Software Issue
00 to 99

7.0 808/847 INSTRUMENTS

7.1 808/847 Mnemonics

MNEMONICS OF PARAMETERS ACCESSIBLE VIA THE EXTERNAL COMMUNICATIONS LINK.

This is a complete list of all parameters available.

ASCII/ HEX	MNEMONICS	PARAMETER
48 53	HS	Setpoint high limit
4C 53	LS	Setpoint low limit
31 48	1H	High range limit
31 4C	1L	Low range limit
49 49	II	Instrument type hexadecimal
56 4F	V0	Software verison number hexadecimal
43 49	CI	Configuration information
42 4C	BL	Comms buffer and transmission block length
4D 4E	MN	Mode number
50 56	PV	Last measured value read only
53 50	SP	Setpoint read only
4F 50	OP	Output power in auto read only
53 57	SW	Status word hexadecimal
4F 53	OS	Optional status word
58 53	XS	Extended status word
45 52	ER	Error value
53 4C	SL	Local setpoint 1
48 41	HA	High alarm setpoint
4C 41	LA	Low alarm setpoint
44 41	DA	Deviation alarm setpoint
58 50	XP	Proportional band
54 49	TI	Integral time
54 44	TD	Derivative time
48 42	HB	High cutback
4C 42	LB	Low cutback
43 48	CH	Cycle time 1 (output 1 - heat)
43 43	CC	Cycle time 2 (output 2 - cool)
52 47	RG	Relative cool gain 1
42 50	BP	Sensor break power
48 4F	HO	Heat (output 1) output limit
53 52	SR	Ramp-to-setpoint ramp rate
48 62	Hb	Holdback
4C 63	Lc	Loop counter }
72 31	r1	Ramp rate 1 }
6C 31	l1	Level 1 }
74 31	t1	Dwell 1 } Programme version
72 32	r2	Ramp rate 2 } only
6C 32	l2	Level 2 }
74 32	t2	Dwell 2 }

7.2 Status Words

These are made up of sixteen bits, each one of which represents a particular function on the controller as detailed below. These are Read /Only (R/O), Read/Write (R/W) or Read/Clear (R/C)
The 16 bits are sent as four hexadecimal numbers preceded by ">" to warn the computer that the data is hexadecimal. The format is thus SW = >ABCD. These digits are still transmitted in ASCII code.

TYPE	DIGIT	BIT	FUNCTION	ATTRIBUTE	CLEAR/SET
Reserved	D	0	Data Format	R/W	Free/Fixed
	D	1	Sensor Break	R/O	No/Yes
	D	2	Key Disable	R/W	No/Yes
	D	3	Spare	-	-
Alarm 3 and key Handling	C	4	Spare	-	-
	C	5	Parameter Change via keys	R/W	No/Yes
	C	6	Dev. Alm State	R/O	Off/On
	C	7	Dev. Alm Cause	R/O	Absent/Present
Alarms 1 and 2	B	8	Low Alm. State	R/O	Off/On
	B	9	Low Alm. Caus	R/O	Absent/Present
	B	10	Hgh Alm. State	R/O	Off/On
	B	11	Hgh Alm. Cause	R/O	Absent/Present
Command	A	12	Alarm Acknowledge	R/C	ACK/Alarm
	A	13	Spare	-	-
	A	14	Spare	-	-
	A	15	Auto/Manual	R/W	Auto/Manual

Notes:

Alarm State : indicates whether the 808/847 instrument has indentified and flagged an alarm condition.

Alarm Cause : identifies if the alarm condition is still present.

Alarm Ack : writing a logical bit '0' into this bit will cancel the state of any of the alarms, but it will not clear the cause of any of the alarms.

Parameter change : indicates that a parameter has been changed via the front keys. This bit can only be via Keys cleared by the external communications link by writing a logical '0' into this bit location.

Sensor Break : the Sensor Break bit (SW Digit bit no. 1) indicates either a sensor break condition (overrange) or a sensor underrange condition (underrange).

THE OPTIONAL STATUS WORD

These are made up of sixteen bits, each one of which represents a particular function on the controller as detailed below. These are Read /Only (R/O), Read/Write (R/W) or Read/Clear (R/C)
The 16 bits are sent as four hexadecimal numbers preceded by ">" to warn the computer that the data is hexadecimal. The format is thus SW = >ABCD. These digits are still transmitted in ASCII code.

TYPE	DIGIT	BIT	FUNCTION	ATTRIBUTE	CLEAR/SET
Program Status	D	0	Hold Active	R/W	-
	D	1	Ramp Active	R/W	-
	D	2	Spare	-	-
	D	3	Spare	-	-
Current Function	C	4	Spare	-	-
	C	5	Spare	-	-
	C	6	Ramp/Dwell	R/O	Ramp/Dwell
	C	7	Spare	-	-
Current Segment	B	8	Segment	R/O	-
	B	9	Segment	R/O	-
	B	10	Segment	R/O	-
	B	11	Segment	R/O	-
Spares	A	12	Spare	-	-
	A	13	Spare	-	-
	A	14	Spare	-	-
	A	15	Spare	-	-

THE EXTENDED STATUS WORD (XS) in the format (.ABCD)

TYPE	DIGIT	BIT	FUNCTION	ATTRIBUTE	CLEAR/SET
Self Tune	D D D D	0 1 2 3	Self Tune Spare Spare Spare	R/W - - -	Off/On - - -
Spaes	C C C C	4 5 6 7	Spare Spare Spare Spare	- - - -	- - - -
Spares	B B B B	8 9 10 11	Spare Spare Spare Spare	- - - -	- - - -
Spares	A A A A	12 13 14 15	Spare Spare Spare Spare	- - - -	- - - -

7.3 Programme Control on 808 and 847

Programme control is set by the value of D in the Optional Status Word OS in section 7.2. Program Status is a nibble having the value 0 to 5:

D = 0	Reset Programmer/Ramp Function. Read/Write
D = 1	N/A
D = 2	Run Programmer/Ramp Function. Read/Write
D = 3	Hold Programmer. Read/Write
D = 4	N/A
D = 5	RMP Engaged. Read only

Note: in particular bit 5 of OS which may be set to make the programme skip the current segment.
Examples:Writing

OS = >0000	will reset programme.
OS = >0002	will run programme
OS = >0003	will hold programme

7.4 808 and 847 Instrument Identity

Instrument Identity (II)

This will consist of four hex characters in the format >ABCD

ABC = Instrument Type 808,847
D = Software Class
0 to 9 Production Release
A to E Specials
F Prototype

Software Version (V0)

This will consist of four hex characters in the format >ABCD

A = Software Class
0 to 9 Production Release
A to E Specials
F Prototype

B = Software Version
0 to 9

CD = Software Issue
00 to 99

8.0 480 Addressable A/D and D/A Converter

The 480 module is an addressable D to A and A to D Converter which allows the following functions:

a) Output of an Analogue Value in the range 0 to 10 volts on four channels
b) Read in an Analogue Value in the range 0 to 10 volts on four channels

The 'Read' channels return an ASCII string corresponding to the analogue value present at the time the read command was received via a serial communications link.
The 'Write' channels will set an analogue value corresponding to the value transmitted to the unit as an ASCII string via a serial communications link. This value will be maintained until a further write command is sent.

The Serial Link can be RS232C, RS422 or 0-20mA active or passive current loop at 300/1200/2400 or 9600 baud.
The RS422 address is hardware selected to allow identification of up to 256 different modules from address 00 to FF.
Protocol is the same as that of the 820 Series controller with the following mnemonic set.

Mnemonic Set

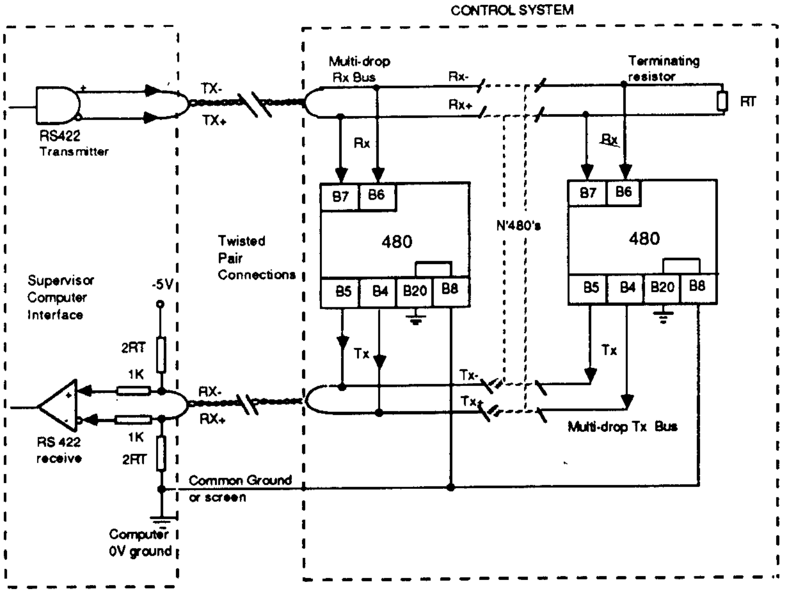
Read-only Channels

ASCII/HEX	MNEMONIC	PARAMETER
52 31	R1	Read the analogue value present at the input to channel 1
52 32	R2	Read the analogue value present at the input to channel 2
52 33	R3	Read the analogue value present at the input to channel 3
52 34	R4	Read the analogue value present at the input to channel 4

Write-only Channels

ASCII/HEX	MNEMONIC	PARAMETER
45 31	E1	Set the analogue value present at the output to channel 1
45 32	E2	Set the analogue value present at the output to channel 2
45 33	E3	Set the analogue value present at the output to channel 3
45 34	E4	Set the analogue value present at the output to channel 4

Values to be set or which are read are sent in engineering units as 4 digits, D1 D2 D3 D4 as in section 2.2 and 3.1 where 9999 corresponds to 9.999 volts, 0123 as 0.123 volts etc.



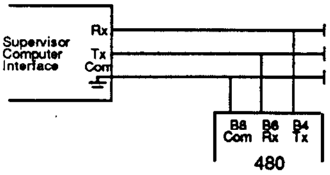


Fig 1 RS422 Connections

Fig 2 RS232 Connections

APPENDIX 1

ASCII CODES

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

HEX-ASCII TABLE complete list

00	NUL	15	NAK	2B	+	40	@	56	V	6B	k
01	SOH	16	SYN	2C	,	41	A	57	W	6C	l
02	STX	17	ETB	2D	-	42	B	58	X	6D	m
03	ETX	18	CAN	2E	.	43	C	59	Y	6E	n
04	EOT	19	EM	2F	/	44	D	5A	Z	6F	o
05	ENQ	1A	SUB	30	0	45	E	5B	[	70	p
06	ACK	1B	ESC	31	1	46	F	5C	/	71	q
07	BEL	1C	FS	32	2	47	G	5D	]	72	r
08	BS	1D	GS	33	3	48	H	5E	^	73	s
09	HT	1E	RS	34	4	49	I	5F	-	74	t
0A	LF	1F	US	35	5	4A	J	60	`	75	u
0B	VT	20	space	36	6	4B	K	61	a	76	v
OC	FF	21	!	37	7	4C	L	62	b	77	w
0D	CR	22	"	38	8	4D	M	63	c	78	x
0E	SO	23	£	39	9	4E	N	64	d	79	y
OF	SI	24	$	3A	:	4F	0	65	e	7A	z
10	DLE	25	%	3B	;	50	P	66	f	7B	{
11	DC1(X-ON)	26	&	3C	<	51	Q	67	g	7C	|
12	DC2	27	'	3D	=	52	R	68	h	7D	}
13	DC3(XOFF)	28	(	3E	>	53	S	69	i	7E	~
14	DC4	29	)	3F	?	54	T	6A	j	7F	DEL
		2A	*			55	U				

APPENDIX 2

EXAMPLES OF COMMUNICATION MESSAGES

1. INSTRUMENT: 820 ADDRESS: 00

(a) Mnemonic: SW **Function: READ**

Computer:	04	30	30	30	30	53	57	05			
	EOT	0	0	0	0	S	W	ENQ			
Controller:	02	53	57	3E	30	30	30	30	03	39	(=BCC)
	STX	S	W	>	0	0	0	0	ETX		

Comment: Data Format is FREE, Controller is in AUTO with LOCAL setpoint. There are no alarms.

(b) Mnemonic: SP **Function: READ**

Computer:	04	30	30	30	30	53	50	05			
	EOT	0	0	0	0	S	P	ENQ			
Controller:	02	53	50	20	20	34	34	2E	03	2E	(=BCC)
	STX	S	P			4	4	●	ETX		

Comment: Setpoint is 44.

(c) Mnemonic: SP **Function: WRITE**

New value : 99

Computer:	04	30	30	30	30	02	53	50	39	39	03	00	=BCC)
	EOT	0	0	0	0	STX	S	P	9	9	ETX		
Controller:	15												
	NAK												

Comment: SP is READ ONLY: use SL:-

(d) Mnemonic: SL **Function: WRITE**

New value: 99

Computer:	04	30	30	30	30	02	53	4C	39	39	03	1C	(=BCC)
	EOT	0	0	0	0	STX	S	L	9	9	ETX		
Controller:	06												
	ACK												

Comment: Setpoint is at last 99.

(e) Mnemonic: OP Function: READ

Computer:	04	30	30	30	30	4F	50	05			
	EOT	0	0	0	0	O	P	ENQ			
Controller:	02	4F	50	20	36	31	2E	39	03	2C	(=BCC)
	STX	O	P		6	1	●	9	ETX		

Comment: Output power is currently 61.9%.

(f) Mnemonic: OP Function: WRITE

New value: 50.0

Computer:	04	30	30	30	30	02	4F	50	35	30	2E	30	03	07 (=BCC)
	EOT	0	0	0	0	STX	O	P	5	0	●	0	ETX	
Controller:	15													
	NAK													

Comment: From SW in (a) above the controller is in AUTO.OP is read only in this condition. Put the controller is MANUAL first:-

(g) Mnemonic: SW Function: WRITE

New value: >8000

Computer:	04	30	30	30	30	02	53	57	3E	38	30	30	30	03	31 (=BCC)
	EOT	0	0	0	0	STX	S	W	>	8	0	0	0	ETX	
Controller:	06														
	ACK														

Comment: The controller is now in MANUAL, the output power can be changed.

(h) Mnemonic: OP Function: WRITE

New value: 25.0

Computer:	04	30	30	30	30	02	32	35	2E	30	03	05	(=BCC)
	EOT	0	0	0	0	STX	2	5	●	0	ETX		
Controller:	06												
	ACK												

Comment: Output power can be changed in manual. Finally return controller to AUTO:-

(I) Mnemonic: SW Function : WRITE

New value :>0000

Computer:	04	30	30	30	30	02	3E	30	30	30	30	03	39	(=BCC)
	EOT	0	0	0	0	STX	>	0	0	0	0	ETX		
Controller:	06													
	ACK													

Comment: Controller now back in AUTO.

2. INSTRUMENT: 822 ADDRESS: 15

(a) Mnemonic: SW Function: READ

Computer:	04	31	31	35	35	53	57	05		
	EOT	1	1	5	5	S	W	ENQ		
Controller:	02	53	57	3E	30	30	30	30	03	39(=BCC)
	STX	S	W	>	0	0	0	0	ETX	

Comment: Programmer is in AUTO; Format is FREE.

(b) Mnemonic: OS Function: READ

Computer:	04	31	31	35	35	4F	53	05			
	EOT	1	1	5	5	O	S	ENQ			
Controller:	02	4F	53	3E	30	30	30	30	03	21	(=BCC)
	STX	O	S	>	0	0	0	0	ETX		

Comment: Programmer is in RESET. A programme may now be selected:-

(c) Mnemonic: CP Function: WRITE

New value: 1

Computer:	04	31	31	35	35	02	43	50	31	03	21 (=BCC)
	EOT	1	1	5	5	STX	C	P	1	ETX	
Controller:	06										
	ACK										

Comment: Selected Programme 1. Programmer is still in RESET. The programme can now be loaded:-

(d) Mnemonic: OS Function: WRITE

New value:>0001

Computer:	04	31	31	35	35	02	4F	53	3E	30	30	30	31	03	20 (=BCC)
	EOT	1	1	5	5	STX	O	S	>	0	0	0	1	ETX	
Controller:	06														
	ACK														

Comment: Programme 1 is now loaded ie. at LEVEL 0 (or SERVO). It can now be started:-

(e) Mnemonic: OS Function: WRITE

New value:>0002

```
Computer:04  31   31   35   35   02   4F   53   3E   30   30   30   32  03   23 (=BCC)
        EOT  1    1    5    5    STX  O    S    >    0    0    0    2   ETX
```

```
Controller:  06
             ACK
```

Comment: Programme has now started. Note that the previous example OS=>0001 could have been ommitted as >0002 would have started the programme without going through the "load" stage. The current segment can be read:-

(f) Mnemonic: CS Function: READ

```
Computer:    04   31   31   35   35   43   53   05
             EOT  1    1    5    5    C    S    ENQ
```

```
Controller:  02  43   53   20   20   20   31   2E   03   2C   (=BCC)
             STX C    S                   1    •    ETX
```

Comment: Programme is in segment 1. This can be changed:-

(g) Mnemonic: CS Function: WRITE

New value : 3

```
Computer:    04   31   31   35   35   02   43   53   33   2E   03   0E   (=BCC)
             EOT  1    1    5    5    STX  C    S    3    •    ETX
```

```
Controller:  15
             NAK
```

Comment: CS can only increase the segment number by 1, ie. from 1 to 2:-

(h) Mnemonic: CS Function: WRITE

New value: 2

```
Computer:    04   31   31   35   35   02   43   53   32   03   21   (=BCC)
             EOT  1    1    5    5    STX  C    S    2    ETX
```

```
Controller:  06
             ACK
```

Comment: Programmer is now in segment 2. Using CS=3 etc the programme can be stepped through segment by segment. Alternatively OS=>0004 will force the programme to the END state, OS=>0003 will put it in HOLD, OS=>0002 to continue or OS=>0000 to RESET when CP can be used to select another (existing) programme.

Finally with the programme still running the Setpoint can be inspected:-

(i) Mnemonic : sp Function: READ

```
Computer:      04    31    31    35    35    73    70    05
               EOT   1     1     5     5     s     p     ENQ

Controller:       02    73    70    04
                  STX   s     p     EOT
```

Comment: This Mnemonic is UPPER CASE! Notice how the controller throws back the unrecognized mnemonics. Try again:-

(j) Mnemonic: SP Function: READ

```
Computer:      04    31    31    35    35    53    50    05
               EOT   1     1     5     5     S     P     ENQ

Controller:    02    53    50    20    31    35    30    2E    03    3A   (=BCC)
               STX   S     P           1     5     0     ●     ETX
```

Comment: Setpoint is 150.

APPENDIX 3

SAMPLE PROGRAMS

Note:

1. These programs are written in GW Basic and are BASICA compatible. They will run on IBM and compatibles. They are available on $5^{1}{}_{4}$ in floppy disc - Eurotherm Part No: DISC 1 IBM

2. Wiring connections for digital communications, RS232, between an 800 series instrument and an IBM Personal Computer:

Instrument rear terminal connections:

	808/847	815/818	820	821	822	825
RX	15	13	14	14	14	14
TX	13	15	16	16	16	16
COM	11	17	12	12	12	12

IBM PC/XT & OLIVETTI	
Pin no.	
2	Rx (800 inst Tx)
3	Tx (800 inst Rx)
7	Com
4 5	Link together
6 8 20	Link together

IBM AT	
Pin no.	
2	Rx (800 inst Tx)
3	Tx (800 inst Rx)
5	Com
7 8	Link together
4 1 6	Link together

PARAMETER READ PROGRAM

Program : COMMS.BAS
Version : A.2

This simple program reads 12 parameters stored in the instrument and displays them on the computer screen. The mnemonics for the 12 parameters are listed on lines 460 to 570 and these may be altered as necessary. If less or more parameters are required then change the variable LASTPARAM in line 440 accordingly.

The program communicates via COM1 at 9600 baud. If either of these need to be changed then alter the statement in line 640. Possible baud rates are 300, 600, 1200, 1800, 2400, 4800 and 9600. Changing COM1 to COM2 will communicate via the second RS232 port if fitted.

```
1Ø REM PROGRAM : COMMS.BAS
2Ø REM VERSION : A.2
3Ø REM DATE    : November 1986
4Ø REM ------------------------------------------------------------
5Ø REM INITIALISE SCREEN
6Ø REM ------------------------------------------------------------
7Ø CLS                               ' Clear the screen for the first time only
8Ø CYCLE = Ø                         ' Count how many cycles we do
9Ø GOSUB 34Ø                         ' Setup global definitions
1ØØ GQSUB 61Ø                        ' Setup COMMS port
11Ø LOCATE 2, 29                    ' Display program title at top of screen
12Ø PRINT "C O M M S  D E M O"
13Ø REM -----------------------------------------------------------
14Ø REM MAIN LOOP
15Ø REM -----------------------------------------------------------
16Ø CYCLE = CYCLE + 1                 ' Next cycle
17Ø LOCATE 4,32
18Ø PRINT "CYCLE : ";CYCLE;
19Ø LNUMB=6                          ' Starting line for printing parameter value
2ØØ FOR PN = 1 TO LASTPARAM
21Ø   PARAMETER$ = PNAME$(PN)        ' Get mnemonics for next parameter
22Ø   LOCATE LNUMB, 31
23Ø   PRINT ">";
24Ø   GOSUB 67Ø                      ' Build Packet
25Ø   GOSUB 72Ø                      ' Send packet and wait for unit to reply
26Ø   LOCATE LNUMB, 32               ' Position cursor to next line on display
27Ø   PRINT PNAME$(PN);" = ";RX$;"                ";
28Ø   LOCATE LNUMB, 31
29Ø   PRINT " ";
3ØØ   LNUMB = LNUMB + 1              ' Move to next line
31Ø NEXT PN
32Ø IF CYCLE > 1ØØØ THEN CYCLE = Ø ' Reset after doing this many
33Ø GOTO 16Ø                        ' Repeat main loop for ever
34Ø REM -----------------------------------------------------------
35Ø REM INITIALISE PARAMETERS
36Ø REM -----------------------------------------------------------
37Ø ADDRESS$ = "ØØØØ"               ' X.Y Address of unit to be interrogated
38Ø STX$ = CHR$(&H2)                ' Protocol for comms packets
39Ø ETX$ = CHR$(&H3)
4ØØ EOT$ = CHR$(&H4)
41Ø ENQ$ = CHR$(&H5)
42Ø ACK$ = CHR$(&H6)
43Ø NAK$ = CHR$(&H15)
44Ø LASTPARAM = 12                  ' Last parameter we are to look at
45Ø DIM PNAME$( LASTPARAM )         ' Define parameter Mnemonics
46Ø PNAME$(1) = "PV"
47Ø PNAME$(2) = "SP"
48Ø PNAME$(3) = "OP"
49Ø PNAME$(4) = "SL"
5ØØ PNAME$(5) = "1A"
51Ø PNAME$(6) = "2A"
52Ø PNAME$(7) = "HS"
53Ø PNAME$(8) = "LS"
54Ø PNAME$(9) = "XP"
55Ø PNAME$(1Ø) = "TI"
56Ø PNAME$(11) = "TD"
57Ø PNAME$(12) = "SW"
58Ø TRUE = Ø
59Ø FALSE = -1
6ØØ RETURN
```

```
610 REM ------------------------------------------------------------
620 REM SETUP_COMMS_PORT
630 REM ------------------------------------------------------------
640 COMFIL$ = "COM1:9600,E,7,1"    ' Set BAUD rate here
650 OPEN COMFIL$ AS £1
660 RETURN
670 REM ------------------------------------------------------------
680 REM BUILD_PACKET
690 REM ------------------------------------------------------------
700 PACKET$ = EOT$ + ADDRESS$ + PARAMETER$ + ENQ$
710 RETURN
720 REM ------------------------------------------------------------
730 REM SEND_PACKET
740 REM ------------------------------------------------------------
750 IF NOT EOF(1) THEN RX$ = INPUT$(LOC(1), £1)
760 RX$ = ""
770 IN$ = ""
780 PRINT £1,PACKET$;
790 INTERVAL = 1000                    ' How long to wait before we timeout
800 TIM = 0                            ' Clear timeout counter
810 TIMEOUT = FALSE                    ' Clear timeout flag
820 RXEND = FALSE                      ' Clear recieve completed flag
830 WHILE (RXEND=FALSE) AND (TIMEOUT=FALSE) ' Terminate on rxend or timeout
840   WHILE NOT EOF(1)                 ' If comms buffer contains data then get it
850     IN$ = INPUT$( 1, £1 )          ' Read next character from comms buffer
860     RX$ = RX$ + IN$                ' Append to received data string
870     IF (IN$=ETX$) OR (IN$=ACK$) OR (IN$=NAK$) THEN RXEND = TRUE
880     TIM = 0                        ' Reset timeout if something was received
890   WEND                             ' Buffer empty
900   TIM = TIM + 1                    ' May have to timeout'if we dont get it soon
910   IF (TIM > INTERVAL) THEN TIMEOUT = TRUE
920 WEND                               ' rxend or timeout
930 IF LEN(RX$) <= 1 THEN 1060
940 BCCERROR = FALSE
950 BCC = 0
960 FOR  X = 2 TO (LEN( RX$ )-1)
970   BCC = BCC XOR ASC(MID$(RX$,X,1))
980 NEXT X
990 IF CHR$(BCC) <> IN$ THEN BCCERROR = TRUE ' last character in in$ was BCC
1000 IF BCCERROR = TRUE THEN RX$ = "BCC ERROR": GOTO 1090
1010 SOD = INSTR( RX$, STX$ ) + 3           ' find first data character
1020 EOD = INSTR( RX$, ETX$ )               ' find last data character
1030 LOD = EOD - SOD                        ' length of data
1040 RX$ = MID$( RX$, SOD, LOD )
1050 GOTO 1090
1060 IF TIMEOUT = TRUE THEN RX$ = "TIMEOUT"
1070 IF RX$ = ACK$ THEN RX$ = "ACK"
1080 IF RX$ = NAK$ THEN RX$ = "NAK"
1090 RETURN
```

PARAMETER WRITE PROGRAM

Program : COMMS2WR.BAS
Version : A.2

This simple program allows alteration to be made to the commissioning parameters in Eurotherm communicating controllers. The user is prompted for a parameter, which must be the appropriate two digit mnemonic. The user is then prompted for a new value for this parameter. The data is constructed into a string and sent to the unit. The mnemonic, data and reply string are then shown on the computer screen.

The baud rate can be changed altering the command in line 550. Possible baud rates are 300, 600, 1200, 1800, 2400, 4800, and 9600.

```
10 REM PROGRAM : COMMSWR.BAS
20 REM VERSION : A.2
30 REM DATE    : November 1986
40 REM ------------------------------------------------------------------
50 REM INITIALISE VARIABLES
60 REM ------------------------------------------------------------------
70 GOSUB 420                     ' Setup global definitions
80 GOSUB 550                     ' Setup COMMS port
90 REM ------------------------------------------------------------------
100 REM DRAW MAIN SCREEN
110 REM ------------------------------------------------------------------
120 CLS
130 LOCATE 2,24                    ' Display program title at top of screen
140 PRINT "C O M M S  W R I T E  D E M O"
150 LOCATE 12,28                   ' Parameter mnemonic
160 PRINT "Parameter         : ";
170 LOCATE 13,28                   ' Value
180 PRINT "Value             : ";
190 LOCATE 14,28                   ' 820 Responses
200 PRINT "Message Received  : ";
210 REM ------------------------------------------------------------------
220 REM MAIN LOOP
230 REM ------------------------------------------------------------------
240 PROMPT$ = "Enter Parameter "            ' Get user to enter parameter name
250 GOSUB 610
260 IF LEN(USERENTRY$) <> 2 THEN 240        ' Ignore blank entries
270 CMD$ = USERENTRY$                       ' Save parameter mnemonic
280 PROMPT$ = "Enter New Value "            ' Get user to enter new value
290 GOSUB 610
300 IF LEN(USERENTRY$) = 0 THEN 240         ' Ignore blank entries
310 VALUE$ = USERENTRY$                     ' Save new value
320 LOCATE 12,48                            ' Report new value on screen
330 PRINT CMD$;"           ";
340 LOCATE 13,48
350 PRINT VALUE$;"           ";
360 GOSUB 710                              ' Change 820 parameter
370 GOTO 240
380 STOP
390 REM --------------------------------------------------------------
400.REM SETUP_GLOBALS
410 REM --------------------------------------------------------------
420 ADDRESS$ = "0000"             ' X.Y Address of 820 to be interrogated
430 STX$ = CHR$(&H2)              ' Protocol for comms packets
440 ETX$ = CHR$(&H3)
450 EOT$ = CHR$(&H4)
460 ENQ$ = CHR$(&H5)
470 ACK$ = CHR$(&H6)
480 NAK$ = CHR$(&H15)
490 TRUE = 0
500 FALSE = -1
510 RETURN
520 REM --------------------------------------------------------------
530 REM SETUP_COMMS_PORT
540 REM --------------------------------------------------------------
550 COMFIL$ = "COM1:9600,E,7,1"    ' Set BAUD rate here
560 OPEN COMFIL$ AS £1
570 RETURN
580 REM --------------------------------------------------------------
590 REM USER_ENTRY
600 REM --------------------------------------------------------------
610 LOCATE 23,1                   ' Position for prompt/entry
620 FOR I=1 TO 79 : PRINT " "; : NEXT I ' Clear any previous prompt first
630 LOCATE 23,1                   ' Now ready to display prompt
640 PRINT PROMPT$;                ' Display it
650 INPUT " : ", USERENTRY$              ' Wait for user input.
660 IF LEN(USERENTRY$) = 0 THEN PRINT CHR$(7); ' Sound bell if none entered.
670 RETURN
```

```
68Ø REM --------------------------------------------------------------
69Ø REM MODIFY PARAMETER
7ØØ REM --------------------------------------------------------------
71Ø PACKET$ = STX$ + CMD$ + VALUE$ + ETX$  ' Construct send packet
72Ø BCC$ = PACKET$                          ' Calculate BCC
73Ø GOSUB 134Ø
74Ø PACKET$ = EOT$ + ADDRESS$ + PACKET$ + CHR$(BCC) ' Packet complete
75Ø MESSAGE$ = PACKET$                     ' Then packet
76Ø GOSUB 111Ø
77Ø GOSUB 82Ø                             ' Catch and report any received data
78Ø RETURN
79Ø REM --------------------------------------------------------------
8ØØ REM RECEIVE DATA FROM UNIT
81Ø REM --------------------------------------------------------------
82Ø CHECKBCC = FALSE              ' Set true if we need to check BCC
83Ø TIMEOUT = FALSE               ' Initialise receive status flags
84Ø RXEND = FALSE                 ' break loop if either of flags goes true
85Ø TIM = Ø                       ' Initialise timeout counter
86Ø RX$ = ""                      ' Clear recieved data string
87Ø IN$ = ""                      ' Clear last character read
88Ø WHILE (RXEND=FALSE) AND (TIMEOUT=FALSE) ' Terminate on rxend or timeout
89Ø   WHILE NOT EOF(1)            ' If comms buffer contains data then get it
9ØØ     IN$ = INPUT$( 1, £1 )     ' Read next character from comms buffer
91Ø     RX$ = RX$ + IN$           ' Append to received data string
92Ø     TIM = Ø                   ' Reset timeout when we receive something
93Ø     IF (IN$=ETX$) THEN CHECKBCC = TRUE ' BCC should be next so check it
94Ø     IF (IN$=ETX$) OR (IN$=ACK$) OR (IN$=NAK$) OR (IN$=EOT$) THEN RXEND = TRU
E
95Ø   WEND                        ' Buffer empty
96Ø   TIM = TIM + 1               ' Update timeout counter
97Ø   IF (TIM > 1ØØ) THEN TIMEOUT = TRUE   ' Timeout after count reaches this
98Ø WEND                          ' Timeout or receive complete
99Ø IF CHECKBCC = FALSE THEN 1Ø4Ø
1ØØØ BCC$ = MID$(RX$,1,(LEN(RX$)-1)) ' Strip off BCC character first
1Ø1Ø GOSUB 134Ø                       ' Generate BCC
1Ø2Ø IF CHR$(BCC) <> IN$ THEN RX$ = "BCC ERROR" ' Say we have an error
1Ø3Ø GOTO 1Ø5Ø
1Ø4Ø IF LEN(RX$) = Ø THEN RX$ = "NONE RECEIVED" ' Nothing received so say so
1Ø5Ø DISPLAY$ = RX$                ' Display received data string
1Ø6Ø GOSUB 116Ø
1Ø7Ø RETURN
1Ø8Ø REM --------------------------------------------------------------
1Ø9Ø REM SEND_MESSAGE
11ØØ REM --------------------------------------------------------------
111Ø PRINT £1,MESSAGE$;
112Ø RETURN
113Ø REM --------------------------------------------------------------
114Ø REM PRINT_MESSAGE
115Ø REM --------------------------------------------------------------
116Ø LOCATE 14, 48                    ' Screen position were we report message
117Ø FOR X = 1 TO LEN(DISPLAY$)
118Ø   CH$ = MID$(DISPLAY$,X,1)      ' Look at next character to be displayed
119Ø   IF (LEN(DISPLAY$)>1) AND (X = LEN(DISPLAY$)) THEN CH$ = "<BCC>" : GOTO 12
7Ø
12ØØ   IF CH$ = STX$ THEN CH$ = "<STX>" : GOTO 127Ø
121Ø   IF CH$ = ETX$ THEN CH$ = "<ETX>" : GOTO 127Ø
122Ø   IF CH$ = EOT$ THEN CH$ = "<EOT>" : GOTO 127Ø
123Ø   IF CH$ = ENQ$ THEN CH$ = "<ENQ>" : GOTO 127Ø
124Ø   IF CH$ = ACK$ THEN CH$ = "<ACK>" : GOTO 127Ø
125Ø   IF CH$ = NAK$ THEN CH$ = "<NAK>" : GOTO 127Ø
126Ø   IF CH$ < " " OR CH$ > "~" THEN CH$ = "<£"+ASC(CH$)+">"
127Ø   PRINT CH$;
128Ø NEXT X                         ' Do each character
129Ø PRINT "      "
13ØØ RETURN
131Ø REM --------------------------------------------------------------
132Ø REM GENERATE_BLOCK_CHECK_DIGIT
133Ø REM --------------------------------------------------------------
134Ø BCC = Ø
135Ø FOR  X = 2 TO LEN( BCC$ )          ' From AFTER STX upto and including ETX
136Ø   BCC = BCC XOR ASC(MID$(BCC$,X,1))
137Ø NEXT X
138Ø RETURN
```

APPENDIX 4

WIRING COMMUNICATION SYSTEMS

1. General

The standard of wiring required depends on the environment in which the controllers are being used.

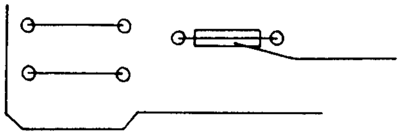
Communication signals are relatively low level and the use of screened twisted pair cables is recommended. Great care is required in the routing of these cables and noisy electrical items such as contactors, solenoids and motors or motor drives must be avoided.

2. Earthing and Insulation

820 Series

The communications of the 820, 821 and 822 are isolated from all other connections on the instrument with the exception of the Digital Inputs and the Digital Outputs.

However, link 1 on the communications pcb connects the common line of the communications board to Earth. Instruments are supplied with this link fitted.



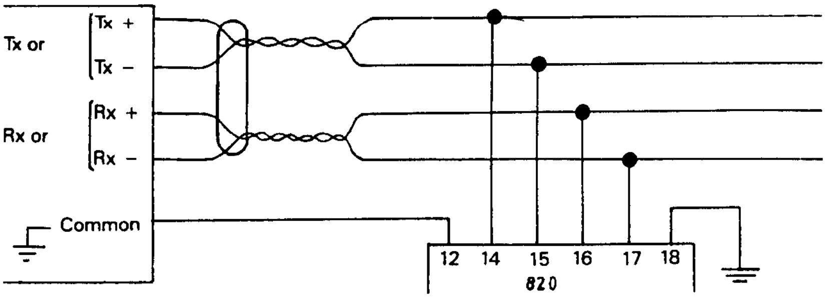
Link 1 on digital Comms Board of 820, 821 or 822

Multiple earth paths should be avoided.

There are two suggested earthing techniques for best common mode rejection.

i) Computer common is earthed. (This is usually the case).
820's are always earthed via terminal 18. With internal link 1 fitted this means the common on the controllers side is also earthed. The mains earth then provides the common return. The screen on the cable should be earthed at one end only - preferably the least 'noisy' end ie. at the computer.

ii) As above but remove internal link 1 from all controllers and connect the screen to pin 12 of all 820's.



This is the preferred technique where computer and controllers are being supplied from completely different supplies.

808/815/818 Series

808, 818 and 815 instruments do not have any internal connections between common and earth, and the communications is isolated from all other connections.

3. RS485 (RS422)

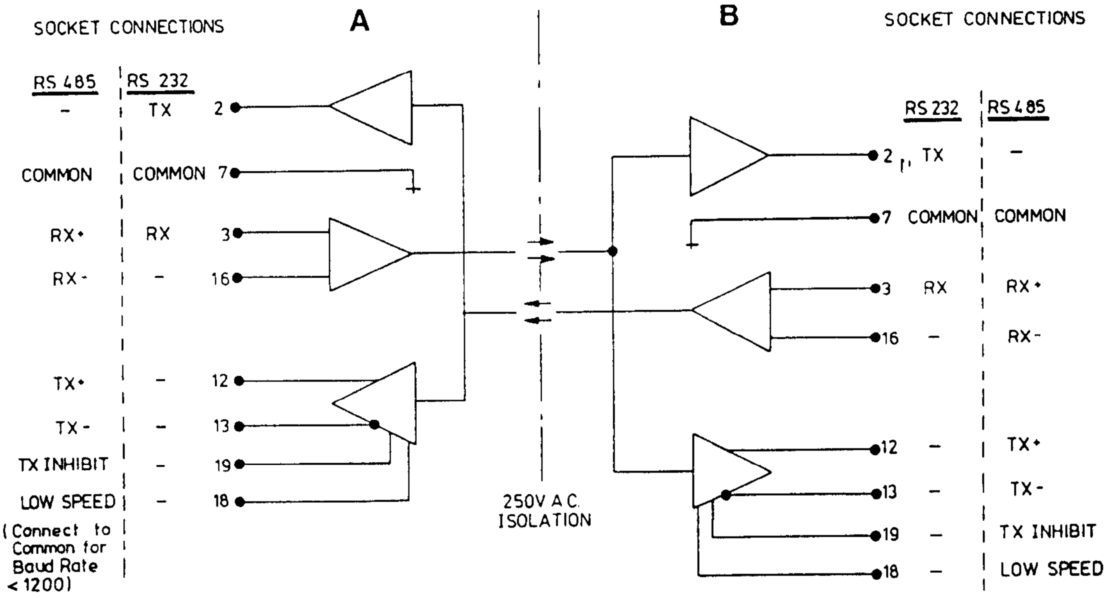
This is the standard for multi-drop applications. RS485 is specified for up to 32 instruments, RS422 to 16.

The transmit lines from the computer/driver should be terminated with a 220 ohm resistor. With the low data rates employed on the 820 it is possible, with short line lengths in some applications, for the line to be not terminated where extra drive capability is required.

Most computers only provide an RS232 port and for multi-drop a RS232-RS422 convertor is required, such as the Eurotherm 261.

4. 261 Connections:

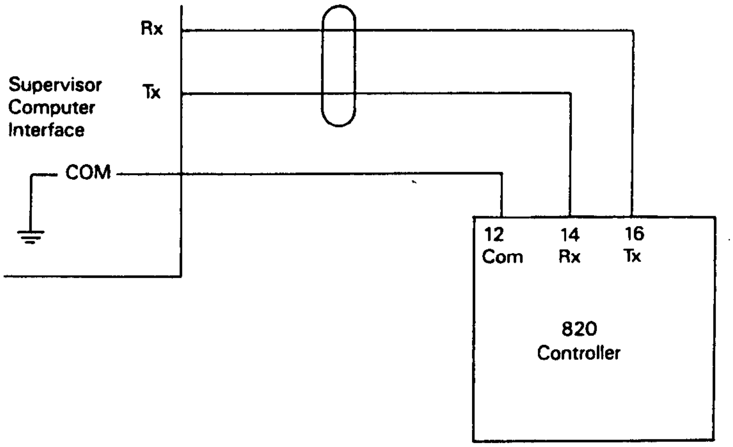
Supply connections: Live - Brown Neutral - Blue Earth - Yellow/Green



Note:
'TX INHIBIT' on RS485 is a facility to enable a computer to place the output driver into the high impedance state. The inhibit is enabled by connection to the Common line. In normal applications it must be left open circuit, in which case it automatically goes high impedance 10msec after the last transmission. (If "LOW SPEED" is enabled the time out is 100msec).

5. RS232 Interface Requirements

The use of biaising and termination resistors are inherent in this standard. RS232 uses an unbalanced line so a common wire is always connected between the 820 and the supervisor. One end of this wire is earthed (most computers have their OV earthed) and can be run through a cable screen to provide added noise immunity for the signal lines. The internal link 1 should be removed from the 820 communications board.



UK REGIONAL SALES AND SERVICE

SOUTHERN AREA

Home Counties, South and
South East
Eurotherm Controls Limited
Faraday Close, Durrington
Worthing
West Sussex BN13 3PL
Telephone Sales: (01903) 695888
Technical: (01903) 695777
Service: (01903) 695444
Fax (01903) 695666
Telex 87114 EUROWG G

MIDLANDS AREA

Midlands, South Wales, East Anglia
and South West
Eurotherm Controls Limited
Miller House
Corporation Street
Rugby Warwickshire CV21 2DW
Telephone (01788) 562011
Fax (01788) 536249

NORTHERN AREA

Northern Counties, North Wales and Scotland
Eurotherm Controls Limited
4/5 Chetham Court
Calver Road, Winwick Quay
Warrington
Cheshire WA2 8RF
Telephone (01925) 572111
Fax (01925) 413099

Sales and support in over 30 countries worldwide

For countries not listed enquiries/orders to:
Eurotherm Controls Limited
Export Dept., Faraday Close, Durrington,
Worthing, West Sussex, BN13 3PL England.
Telephone (01903) 268500
Fax (01903) 265982
Telex 87114 EUROWG G

INTERNATIONAL SALES AND SERVICE

AUSTRALIA
Eurotherm Pty. Ltd.
Telephone Sydney (+61 2) 477 7022
Fax (+61 2) 477 7756

AUSTRIA
Eurotherm GmbH
Telephone Vienna (+43 1) 798 7601
Fax (+43 1) 798 7605
Telex 047 1132000 EIAUT A

BELGIUM
Eurotherm B.V.
Telephone Antwerp (+32 3) 322 3870
Fax (+32 3) 321 7363
Telex 046 33317 EIBNL B

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

FRANCE
Eurotherm Automation SA
Telephone Lyon (+33 78) 664500
Fax (+33 78) 352490
Telex 042 380038 EURTERM F

GERMANY
Eurotherm Regler GmbH
Telephone Limburg (+49 6431) 2980
Fax (+49 6431) 298119
Telex 041 484791 EUROT D
Also regional offices

HONG KONG
Eurotherm Limited
Telephone Hong Kong (+852) 873 3826
Fax (+852) 870 0148
Telex 0802 69257 EIFEL HX

IRELAND
Eurotherm Ireland Limited
Telephone Naas (+353 45) 79937
Fax (+353 45) 75123

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

JAPAN
Eurotherm KK
Telephone Tokyo (+81 3) 33638324
Fax (+81 3) 33638320

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

NETHERLANDS
Eurotherm B.V.
Telephone Leiden (+31 71) 411841
Fax (+31 71) 414526

NEW ZEALAND
Eurotherm Limited
Telephone Auckland (+64 9) 358 8106
Fax (+64 9) 358 1350

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

SPAIN
Eurotherm España SA
Telephone (+34 1) 6616001
Fax (+34 1) 6619093

SWEDEN
Eurotherm AB
Telephone Malmo (+46 40) 435460
Fax (+46 40) 435520

SWITZERLAND
Eurotherm Produkte (Schweiz) AG
Telephone Zurich (+41 1) 8103646
Fax (+41 1) 8108920

UNITED KINGDOM
Eurotherm Controls Limited
Telephone Worthing (+44 1903) 268500
Fax (+44 1903) 265982
Telex 87114 EUROWG G

U.S.A
Eurotherm Controls Inc.
Telephone Reston (+1 703) 471 4870
Fax (+1 703) 787 3436

ED-01

Every effort has been taken to ensure the accuracy of this specification. However in order to maintain our technological lead we are continuously improving our products which could, without notice, result in amendments or omissions to this specification.

Part No. HA020161 Iss 3

Printed in England 0295