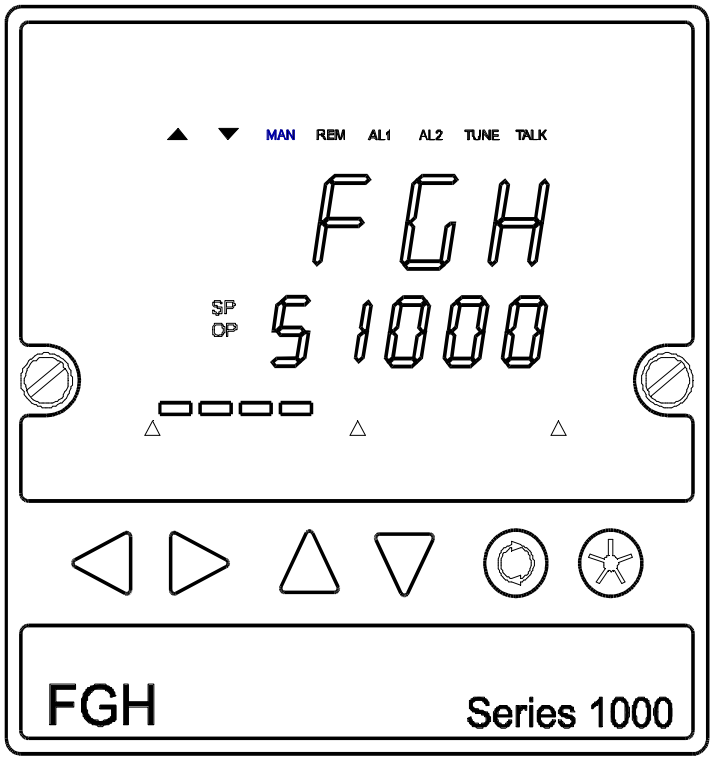
Users Manual



Series 1000 Communications Manual

FGH Controls Ltd
Openshaw Way, Letchworth
Herts, SG6 3ER
Tel: +44(0) 1462 686677
Fax: +44 (0) 1462 671860

1000 COMMUNICATIONS MANUAL
CONTENTS:

1.0 GENERAL DESCRIPTION.

The FGH S1000 controller or P1000 programmer may, if required, be equipped with serial communications. This takes the form of 2 way serial asynchronous communication with a computer.

Messages consist entirely of ASCII characters and may or may not contain spaces as desired.

All messages are terminated with a carriage return, <CR>.

2-Wire EIA-485 (RS-485) or 4-Wire EIA-422-A (RS-422) serial communication standards are user selectable. 3-Wire RS-232 type connection is also available, which will work well in many low specification applications although the performance of the RS-232 type configuration is not guaranteed.

1.1 Specification.

1.1.1 2-Wire EIA-485 Mode (RS485)

(Balanced digital multipoint communication system).

Transmission standard:	EIA-485 (RS-485)
Data rates:	1200, 2400, 4800 and 9600baud.
Data format:	1 start, 7 data, odd parity, 1 or 2 stop bits.
Implementation:	4 wire Full duplex or 2 wire half duplex.
Max drivers per line:	32
Max receivers per line:	32
Max cable length:	1200 metres/3937 feet

1.1.2 4-Wire EIA-422-A Mode (RS422)

(Balanced voltage digital communication system).

Transmission standard:	EIA-422-A (RS-422)
Data rates	1200, 2400, 4800 and 9600 baud.
Data format	1 start, 7 data, odd parity, 1 or 2 stop bits.
Implementation	4 wire full duplex.
Max drivers per line	1
Max receivers per line	10
Max cable length	1200 metres/3937 feet

1.1.3 3-Wire'RS-232'Type

(Unbalanced voltage digital communication system).

Important Note!

This is not a full implementation of RS-232. The output signal is limited to positive voltage levels, although pseudo bipolar output may be obtained by connecting TX- to COM on the controller. *While this will work in many undemanding applications, it's performance is not guaranteed.* The following is given for information only.

Transmission type:	'RS-232' type
Data rates	1200, 2400, 4800 and 9600 baud.
Data format	1 start, 7 data, odd parity, 1 or 2 stop bits.
Implementation	3 wire full duplex
Max drivers per line	1
Max receivers per line	1
Max cable length	15 metres/50 feet

2.0 HARDWARE CONFIGURATION.

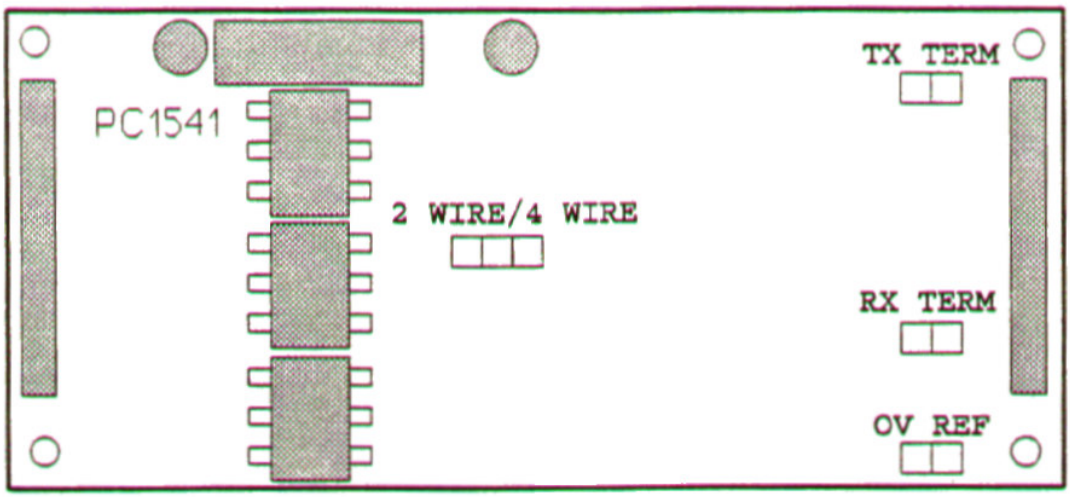
2.1 Shorting Links.

The following links are to be found on the comms board PC1541 inside the instrument and should be set as indicated.

PC 1541 Link	EIA-485	E1A-422-A	'RS-232' Type
TX TERM	FIT*	OPTIONAL	FIT
RX TERM	REMOVE	OPTIONAL	REMOVE
OV REF	OPTIONAL **	OPTIONAL **	OPTIONAL **
4 WIRE/2 WIRE	2 WIRE	4 WIRE	4 WIRE

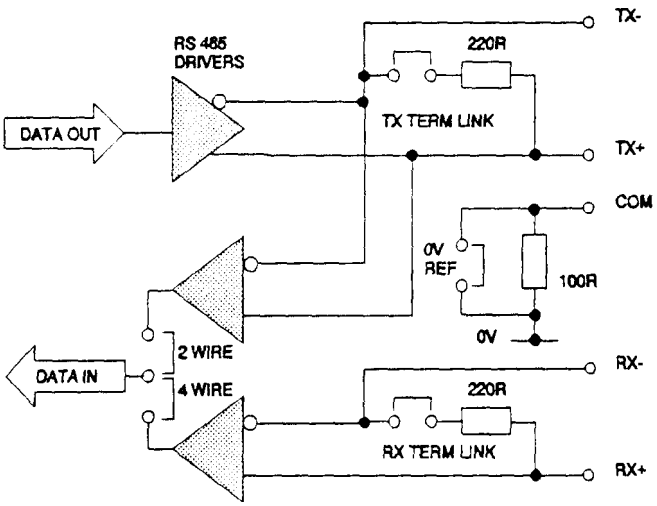
Note * See para 2.3
** See para 2.4

Position of links on PC1541 Communications Board



2.2 Shorting Link Functions

The purpose of the various shorting links is described in the followingparagraphs, but this illustration , being an electrical schematic, may be of use in further understanding their function.



2.3 PC1541 Termination links

If the instrument is being used in EIA-485 mode then up to 32 instruments can be connected together in a multipoint system as a 'daisy chain'. Link TX TERM, the transmission termination resistor should only be fitted to the ***last*** instrument at each end of the chain. When fitted it connects a 220 ohm resistor between the TX+ and TX- lines.

If the instrument is being used in RS 232 type mode then link TX TERM should always be fitted.

2.4 PC-1541 Ground links

A 100 ohm resistor is fitted between COM and the instrument Ov. this can be modified by fitting link OV REF which will then connect COM directly with the instrument Ov. This is provided so that if the instrument earth is not connected to the same earth as the other items of communications equipment, then a third wire could be used to provide a signal return. In this situation the link OV REF should be removed.

When the same Earth ***is*** connected to Frame Ground then link OV REF should be fitted and a 100 ohm 1/2 watt resistor fitted between terminal 34 (COM) and terminal 36 (EARTH). This resistor must be of such a type as to become an open circuit when overloaded.

2.5 PC 1541 4-Wire/2-Wire link

If the instrument is to be used in 2-wire RS-485 mode then the 4 WIRE/2 WIRE link should be fitted in position 2 WIRE. This causes the instrument to take it's serial input from terminals 30 and 31 rather than 32 and 33. In 4-wire RS-485 mode, RS-422 mode or 'RS-232' type mode then the 4 WIRE/2 WIRE link should be fitted in position 4 WIRE. This will cause the serial input to be taken from terminals 32 and 33. Serial output from the instrument is always from terminals 30 and 31.

3.0 INSTALLATION

3.1 General

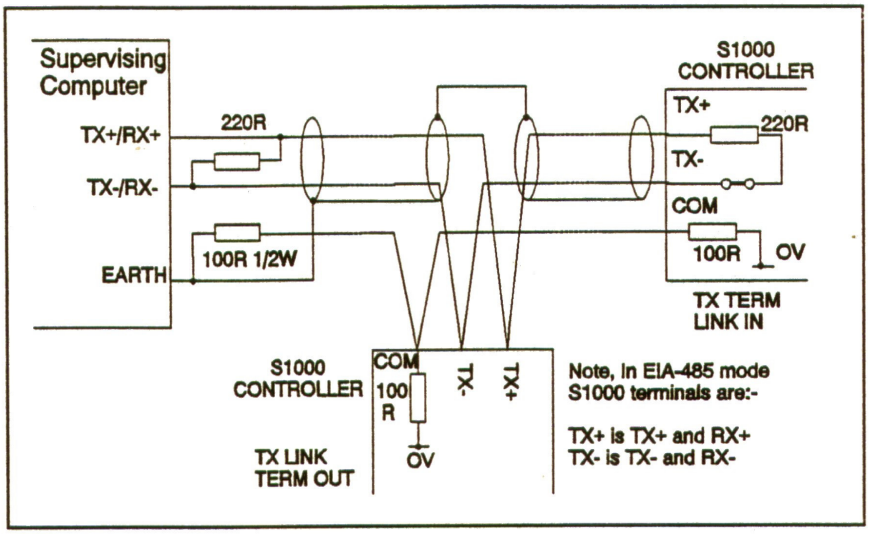
Although simple point to point wiring of communications controller to instruments may work in some cases, this may cause errors if the environment ever worsens.

In order to meet the full specification it may be necessary to connect the system as shown in the following examples.

The use of shielded interconnecting cable is recommended to reduce noise pick-up over long distances or in noisy environments. Under these circumstances the screen should be connected to Frame Ground at either, or in rare cases, at both ends as required.

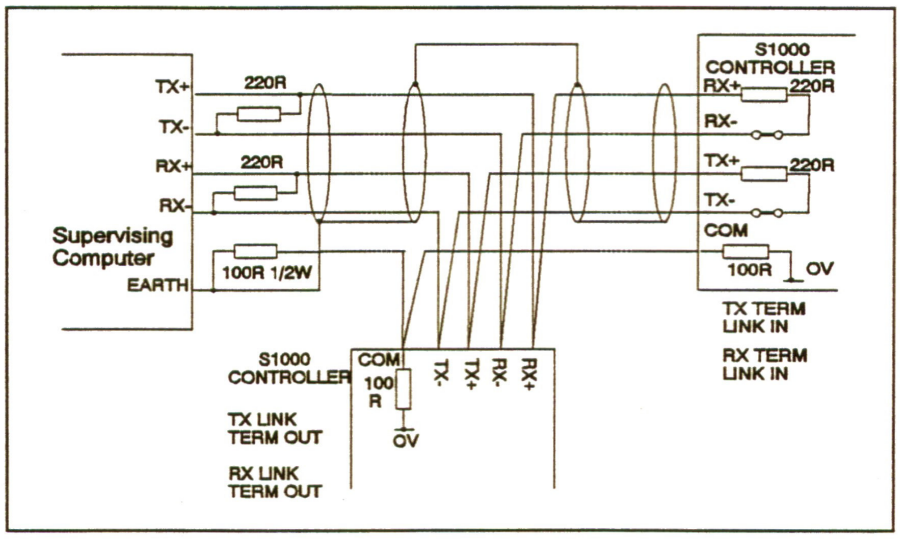
The connections to the rear terminals of the instrument are as follows.

3.2 2 Wire E1A-485 Mode (RS-485)



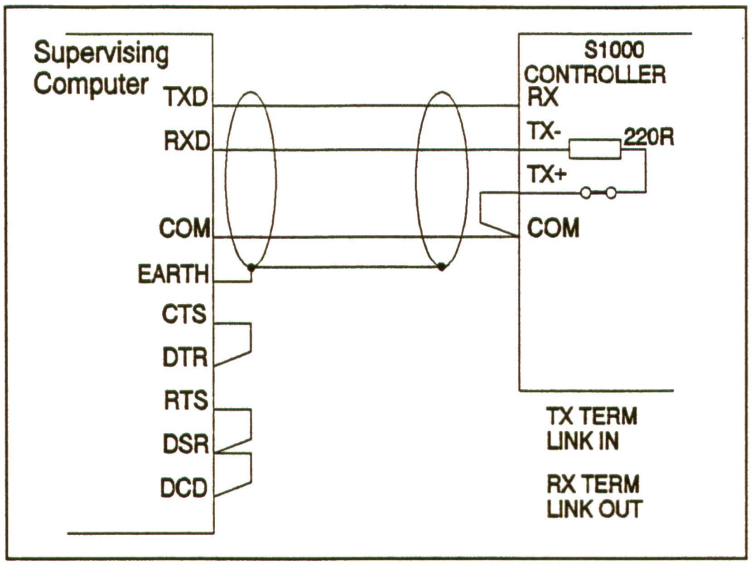
This diagram shows the full implementation connection for EIA-485. It includes the third (COM) wire. This is specified by EIA standard 485 for correct operation, but in practice it is not usually required and is shown here only to conform to the EIA standard. Also shown is a screen around the data lines. This is used to reduce the effect of electrical noise and may be required with long runs or in very noisy environments.

3.3 4-Wire EIA-422-A (RS-422)



This diagram shows the full implementation wiring, this time for EIA-422. It contains a fifth wire (COM) declared in the EIA specification to be necessary for correct operation and so it is included in this diagram. In practice, however, this fifth wire is usually found to be unnecessary. It may be that in very demanding applications the fitting of the (COM) wire and screening assists trouble free operation.

3.4 3-Wire 'RS-232' Type



Shown here is one S1000 implementation of the type of controller connected to a computer by an RS-232 this interface. The use of 'standard' is not recommended as it is not possible to specify which connections will work without knowing exact hardware details, and even then it is difficult. The connections shown here were found to work with PC look-a-likes, but because of the variation in RS-232 implementations, if this does not work then it is recommended that a reference book on RS-232 installation be obtained and used, or better still, an RS-485 interface be obtained for the supervising computer so that the full potential of the S1000/P1000 communications interface may be realised.

3.5 Connection reference table

Instrument Terminal	Function	EIA-485	EIA-422-A	'RS-232' Type
30	TX-	TX-/RX-	TX-	TX
31	TX+	TX+/RX+	TX+	(COM1)
32	RX-	N/C	RX-	RX
33	RX+	N/C	RX+	N/C
34	COM	N/C or SCREEN	N/C or SCREEN	COM (COM2)
1/36	EARTH	N/C or SCREEN	N/C or SCREEN	N/C or SCREEN

3.6 Software Configuration.

Software configuration of the instrument may be performed once the instrument has been placed in 'E mode'. This is done by removing the instrument from its sleeve and fitting a link to the two pins marked EMODE on the CPU board, the instrument is then replaced in its sleeve and power re-applied. Use the scroll button (>) to scroll through elements until the desired one is reached.

Ensure that element **SLOt6** is **'on'** as this informs the instrument that the comms board is fitted in slot 6.

Scroll to the element **bAUd** and select the baud rate required from 1200, 2400, 4800 or 9600 using the up/down keys.

Scroll to the element **AddrS.** This is the controller instrument address and may be set between 0 and 99 using the up/down keys. If the instrument is a P1000 programmer then the controller part of the instrument will have an address as set by this parameter and the programmer part of the instrument will have an address which is the controller address plus 16.

4.0 CONTROLLER COMMUNICATIONS PROTOCOL

Messages to and from the S1000 controller or controller section of the P1000 vary in form depending on, amongst other things, the type of message and its contents.

4.1.1 Write messages

Write messages to the controller take the following form.

WAAPDDDD or WAAP-DDD

where	W	= write message header
	AA	= instrument address
	P	= parameter code
	DDDD	= numeric data (preceded by – if negative)

messages to the controller may have the fields separated by spaces. These spaces will be ignored. Messages for the S1000 will not contain spaces.

Eg.

W 45 C 0123 <CR>

will attempt to write 123 to to the local set point as well as

W45C0123 <CR>

The <CR> at the end of the message is a carriage return. (Hex 0D). Each message written to the instrument must be terminated with, and each message from the instrument will be terminated with a carriage return.

4.1.2 Read messages

Read messages to the controller take the following form.

RAAP
where R = read message header
AA = instrument address
P = parameter code to be read from

Again the message must be terminated with a <CR>, and spaces may be included if desired.

4.1.3 Set instrument status messages

Set messages to the controller take the following form.

SAAC
where S = set message header
AA = instrument address
C = set parameter code to be performed

Again the message must be terminated with a <CR>, and spaces may be included if desired.

4.2 Message header.

This may be;

ASCII R signifying a Read
ASCII W signifying a Write
ASCII S signifying a Set.

The R header is used whenever data is being read from the S1000. When this header is used the data field is absent. The W header is used to write data to the S1000. The S header is used to set the status of the S1000.

4.3 Controller address.

Each instrument must be given a unique address between 0 and 99. This is set when the instrument is in E mode. The address field of the message, consisting of two ASCII characters determines to which instrument the message is directed. The S1000 ignores the message unless it corresponds to its own address.

4.3.1 Wildcard addresses

If desired, a group of instruments can be written to together by using a wildcard character (large X) in place of one or more of the address characters.
E.g.

W6XC0100 <CR>

would result in all instruments on that communication line with addresses of 60 to 69 having their local set point set to 100.

Instruments written to with wildcard addresses do not reply.

Please note : Wildcard addressing is not applicable to the P1000.

4.4 Message parameter code.

The parameter code field of the message is a single ASCII character. In a write message this corresponds to one of the control parameters listed in paragraph 4.6, 'parameter codes'. In the case of set messages, this is still a single ASCII character, but corresponds to one of the 'set status codes' listed in paragraph 4.7.

4.5 Message data field

The message data field consists of four ASCII characters, and may be preceded by a 'minus sign', if the data is negative. These characters represent the decimal value of that parameter in stored units. That is, if the parameter were specified as being in units of 0.1% and read 0110 then this would indicate a reading of 11%. Note that there must be four digits in the data field, if necessary use zeros to pad out a small data number.

4.6 Controller read/write parameter codes

each parameter within the S1000 is assigned a single alpha code. The meaning of this code may vary according to the controller action type, eg. heat/cool, motorised valve etc. Some parameters are read only, these are indicated by *.

@	COMMS REMOTE SET POINT
A*	MEASURED VALUE
B	OUTPUT IN 0.1%
	DESIRED VALVE POSITION IN 0.1%
C	LOCAL SET POINT
D	PROPBAND IN 0.1%
	RATIO IN 0.1%
E	INTEGRAL ACTION TIME IN SEC
	RATIO MINIMUM OUTPUT LIMIT (th-lo)
F	DERIVATIVE ACTION TIME IN SECONDS
	RATIO NEGATIVE THERMAL HEAD LIMIT (l-op)
G	APPROACH BAND IN 0.1%
H	UPPER POWER LIMIT IN %
	RATIO MAXIMUM OUTPUT LIMIT (h-op)
I	CYCLE TIME IN SECONDS
	CYCLE TIME (HEAT) IN SECONDS
	RATIO LIMIT 1 REFERENCE (ref-h)
J	ALARM 1 LEVEL
K	ALARM 2 LEVEL
L*	CONTROLLER STATUS
M	INTEGRAL APPROACH BAND
N*	RESULTANT SET POINT
O	SET POINT TYPE CODE
P	ALARM 1 TYPE CODE
Q*	INSTRUMENT TYPE CODE
R*	ANALOGUE REMOTE SET POINT ^
S	ALARM 2 TYPE CODE
T	HEAT ONLY LOW POWER LIMIT
	COOL POWER LIMIT
	RATIO MAXIMUM THERMAL HEAD (th-hi)
U	RATE OF CHANGE OF SET POINT
V	CYCLE TIME (COOL) IN SECONDS
W	COOL RELATIVE PROPBAND IN TENTHS
X	HEAT/COOL DEADBAND
	MOTORISED VALVE DEADBAND
Y	AUXILIARY SET POINT 1
Z	AUXILIARY SET POINT 2

4.7 Controller Set status codes.

Writing a set command to the instrument with a parameter as follows will produce the specified action if the address field matches the address of the instrument.

M SET CONTROLLER TO MANUAL
A SET CONTROLLER TO AUTO
P TURN ON PRETUNE
T TURN ON ADAPTIVE TUNE
0 TURN OFF PRETUNE AND ADAPTIVE TUNE
U UNLATCH LATCHED ALARMS

4.8 Response from read or write

the response of the controller to a satisfactory read or write message with the correct address will be as follows (unless an address wildcard is used, see para 4.3.1):

*AAPDDDD or *AAP-DDDD

The controller will respond with a string of ASCII characters. The header will consist of '*' (Hex 2A). The header will be followed by an address (AA) showing the address of the responding instrument in ASCII numbers, 00 to 99.

After the address is a single ASCII character showing the parameter selected.

This is followed by a data field of four ASCII numbers representing the decimal value of the parameter selected. This will be preceded by a minus sign '-' if the data is negative.

The units of the data field are the stored units of that parameter. For example, if parameter B were read then the data field units would be in 0.1%, so a reading of 0120 would indicate an output of 12%.

This is followed by a <CR> to complete the message return.

4.8.1 Set status response

The response of the controller to a satisfactory 'set status' message with the correct address will be as follows (unless a wildcard address is used):

*AAP

The controller will respond with a string of ASCII characters. The header will consist of '*' (Hex 2A). The header will be followed by an address (AA) showing the address of the responding instrument in ASCII, 00 to 99.

After the address is a single ASCII character showing the 'set status' mnemonic used This is followed by a <CR> to complete the message return.

4.9 Controller error responses

Two sorts of error in a received message may be detected by the controller, these are:

4.9.1 Corrupt message response

Noise or interference during the transmission of the message causing corruption of one or more characters so that it was no longer valid. The receiver within the instrument detects this, and as long as it was not the address that was corrupted, the controller responds as follows;

?AAC

where AA is the address of the instrument responding
C = P for detected parity error
F for detected overflow error
0 for detected receiver overrun

4.9.2 Syntax error response

Messages that were correctly received but don't make sense, as long as the address part was o. k. generate the following response;

?AANN

where AA = address of the instrument responding
NN = two digit ASCII HEX error code

Error code binary weightings:

bit7 = Illegal trailer
bit6 = Tx buffer overflow
bit5 = Illegal number of characters
bit4 = Illegal data
bit3 = Illegal parameter code
bit2 = Rx buffer overflow
bit1 = Illegal header
bit0 = Write to read only parameter

4.10 SPECIAL PARAMETER CODES

There are several special parameter codes whose data fields are in a coded format.

4.10.1 Instrument status.

The instrument status is available by reading parameter L The instrument will respond with:

*NNU\BCD (RETURN)

Where NN is the instrument address.

A =	Digital inputs	= 0	Both off or unused
		= 1	Input 1 on, 2 off or unused
		= 2	Input 2 on, 1 off or unused
		= 3	Both digital inputs are on.
B ==	Alarm condition	= 0	Both alarms off
		= 1	Alarm 1 on, alarm 2 off
		= 2	Alarm 2 on, alarm 1 off
		= 3	Both alarm 1 and 2 are on
C =	Tuner status	= 0	pretune and atune are off
		= 1	pretune is on, atune is off
		= 2	atune is on, pretune is off
		= 3	pretune and atune are on
D =	Auto/Man	= 0	Automatic
		= 1	Manual

4.10.2 Instrument type code

The instrument type code may be read from the controller by reading parameter code Q. The controller will respond with:

*NNQABCD (RETURN)

Where A = Input 2 = 0 no input 2
= 1 Remote set point board fitted

BC = Input type
= 00 Type S, degrees C
= 01 Type R, degrees C
= 02 Type J, degrees C
= 03 Type K, degrees C
= 04 Type T, degrees C
= 05 Type E, degrees C
= 06 Type B, degrees C
= 07 Type N, degrees C
= 08 Type W, degrees C
= 09 Type W3, degrees C
= 10 Type W5, degrees C
= 11 Type NM, degrees C
= 12 Type L, degrees C
= 13 Type K10, degrees C
= 14 Type T10, degrees C
= 15 Type RT10, degrees C
= 16 Type RT, degrees C
= 17 Type S, degrees F
= 18 Type R, degrees F
= 19 Type J, degrees F
= 20 Type K, degrees F
= 21 Type T, degrees F
= 22 Type E, degrees F
= 23 Type B, degrees F
= 24 Type N, degrees F
= 25 Type W, degrees F
= 26 Type W3, degrees F
= 27 Type W5, degrees F
= 28 Type NM, degrees F
= 29 Type L, degrees F
= 30 Type K10, degrees F
= 31 Type T10, degrees F
= 32 Type RT10, degrees F
= 33 Type RT, degrees F
= 34 Type Linear
= 35 Type Root

D = Control action	0 = None 1 = Heat only 2 = Heat and Cool 3 = Motorised Valve 4 = Ratio output

4.10.3 Alarm Type codes.

Reading and writing the controller alarm types is possible by use of parameters **P** (alarm 1) and **S** (alarm 2) where the data field of the message is coded as follows.

DATA FIELD	S1000 ALARM TYPE	P1000 ALARM TYPE
0000	HIGH ALARM	HIGH ALARM
0001	LOW ALARM	LOW ALARM
0002	INDEXED ALARM	INDEXED ALARM
0003	INDEXED HIGH ALARM	INDEXED HIGH ALARM
0004	INDEXED LOW ALARM	INDEXED LOW ALARM
0005	MANUAL ACK' RELAY	MANUAL ACK' RELAY
0006	REMOTE SP ACK' RELAY	PROGRAM RELAY
0007	INVALID	READY RELAY
0008	INVALID	UP RAMP RELAY
0009	INVALID	DOWN RAMP RELAY
0010	INVALID	SOAK RELAY

4.10.4 Setpoint type codes. (S1000 Only)

The setpoint type (parameter code 0) is coded as follows.

DATA FIELD	SETPOINT TYPE
0000	HIGH CLAMPED SETPOINT
0001	LOW CLAMPED SETPOINT
0002	INDEXED SETPOINT
0003	REMOTE SETPOINT
0004	LOCAL SETPOINT

4.10.5 Ratio limit reference type codes.

The ratio limit 1 reference (parameter code **I**) is coded as follows.

DATA FIELD	RATIO REFERENCE TYPE
0000	LIMIT IS OFF
0001	LOAD TEMPERATURE
0002	SETPOINT TEMPERATURE

5.0 PROGRAMMER COMMUNICATIONS PROTOCOL

Message to the profile generator section of the P1000 vary in form depending on the parameter being accessed. Some parameters require a segment number field to be present and others require an 8 digit data field.

5.1 Programmer addresses.

For serial communications purposes, the programmer and controller parts of the instrument are treated as separate instruments. The controller part will recognise messages whose address field corresponds exactly with the address programmed in engineers mode. The programmer part will recognise the programmed address plus 16. This fact should be borne in mind when assigning controller addresses on the network.

5.2 Message construction.

Messages to the P1000 take the general form.

H AA P SS DDDDDDDD<CR>

Where
- H = message header (R, W or S)
- AA = profile generator address (controller address + 16)
- P = parameter code
- SS = segment number (01 to 25) not always present
- D..D = variable length data field

Messages to the profile generator may contain any number of spaces. Replies from the instrument will not contain spaces.

5.3 Read and write messages.

Read and write messages to the profile generator part of the instrument are slightly more complex than for the controller. Some parameters are applicable to all profiles contained within the instrument, others are applicable only to the profile selected.The profile number of interest is set by a write instruction to the profile pointer (parameter code P) and all following reads and writes will refer to this profile until overwritten by another write to the profile pointer. Segment reads and writes (parameter codes L, R and T) require the SS field to be present in the message. This should be set to the segment number of interest (01 to 25).
As for the controller, a read message is denoted by the R header, and a write message by the W header.

5.4 Programmer read/write parameter codes

The table below shows the meaning of all the parameter codes available in the profile generator.

PARAMETER CODE	SS FIELD	MEANING	READ ONLY	DATA FIELD
C	N/A	Profile setpoint	YES	TYPE 1
D	N/A	Delay start time in mins	NO	TYPE 1
E	N/A	Segment Elapsed time in mins	YES	TYPE 1
H	N/A	Profile hold band in digits	NO	TYPE 1
I	N/A	Profile hold type	NO	TYPE 1 (coded)
J	N/A	Profile repeats	NO	TYPE 1
K	N/A	Number of repeats remaining	YES	TYPE 1
L	YES	Segment target level in digits	NO	TYPE 1
M	N/A	Current event status	YES	TYPE 2
N	N/A	Ready mode event status	NO	TYPE 2
P	N/A	Profile pointer	NO	TYPE 1
Q	N/A	Profile status	YES	TYPES
R	YES	Segment event outputs	NO	TYPE 2

PARAMETER CODE	SS FIELD	MEANING	READ ONLY	DATA FIELD
C	N/A	Profile setpoint	YES	TYPE 1
T	YES	Segment time in minutes	NO	TYPE 4
X	N/A	Profile currently running	YES	TYPE 1

5.5 Data field types.

Due to the wide range of different sorts of parameter that can be accessed on the programmer communications, there are four data field types used.

DATA FIELD TYPE 1

This is identical to the controller data field. It contains four numeric data digits with an optional leading minus sign.

For example, to set the comms profile pointer to 6, on instrument address 4 (remembering that the programmer part is accessed using instrument address +16).

The message **W20P0006<CR>** should be sent.

The instrument would reply. ***20P0006<CR>**

Note. Parameter code **I** (Hold type) uses coded data of the following form.

DATA FIELD	HOLD TYPE
0000	NO INTERNAL HOLD
0005	HOLD ON RAMPS, ABOVE SETPOINT ONLY
0006	HOLD ON RAMPS, BELOW SETPOINT ONLY
0007	HOLD ON RAMPS, ABOVE AND BELOW SETPOINT
0009	HOLD ON DWELLS, ABOVE SETPOINT ONLY
0010	HOLD ON DWELLS, BELOW SETPOINT ONLY
0011	HOLD ON DWELLS, ABOVE AND BELOW SETPOINT
0013	HOLD ON RAMPS & DWELLS, ABOVE SETPOINT ONLY
0014	HOLD ON RAMPS & DWELLS, BELOW SETPOINT ONLY
0015	HOLD ON RAMPS & DWELLS, ABOVE AND BELOW SETPOINT

DATA FIELD TYPE 2

This data field consists of 8 data digits and is used solely for event output data. Each data digit may be either a '1' (signifying event ON), or '0' (signifying event OFF).

For example, To read the current event output status from instrument address 4.

The message **R20M<CR>**
May be answered ***20M10010000<CR>**

Indicating that events 1 and 4 are currently on and the rest are off.

DATA FIELD TYPE 3.

This data field type is used for profile status only, and consists of 4 digits some of which may be alpha characters.

For example, reading the profile status of instrument address 4.

The message **R20Q<CR>**
May invoke the reply ***20QR'dy<CR>**
indicating that the programmer is in ready mode.

Or ***20Q02<CR>**
Indicates that segment 2 is currently running.

Or ***20Q03HM<CR>**
Indicates that segment 3 is running, the H indicates that the programmer is in hold and the M indicates the programmer is recovering from a mains failure.

DATA FIELD TYPE 4

This data field type is only used for parameter code T (segment time). It consists of 4 numeric data digits preceded by an optional alpha character.

For example reading the segment 12 target time from instrument address 4.
(Note the use of the SS field to specify the segment number).

The message **R20T12<CR>**
May invoke the reply ***20T124000<CR>**
Indicating that the segment time is 4000 minutes.

Or ***20T12EOOOO<CR>**
Indicating that the segment is programmed as an END.

Or ***20T12G0008<CR>**
Indicating that the segment is programmed as GOTO program 8.

5.6 Programmer Set commands.

Programmer set commands are signified by the S header as they are for the controller.

There are only four set parameter codes and their meanings are given below.

CODE	MESSAGE	REPLY	MEANING
S	S20S<CR>	*20S<CR>	Start profile pointed to by profile pointer
R	S20R<CR>	*20R<CR>	Reset currently running profile
H	S20H<CR>	*20H<CR>	Holds (pauses) execution of profile
F	S20F<CR>	*20F<CR>	Frees hold allowing profile to continue

5.7 Programmer error responses.

The error responses from the programmer are identical to those used by the controller. See section 4.9.