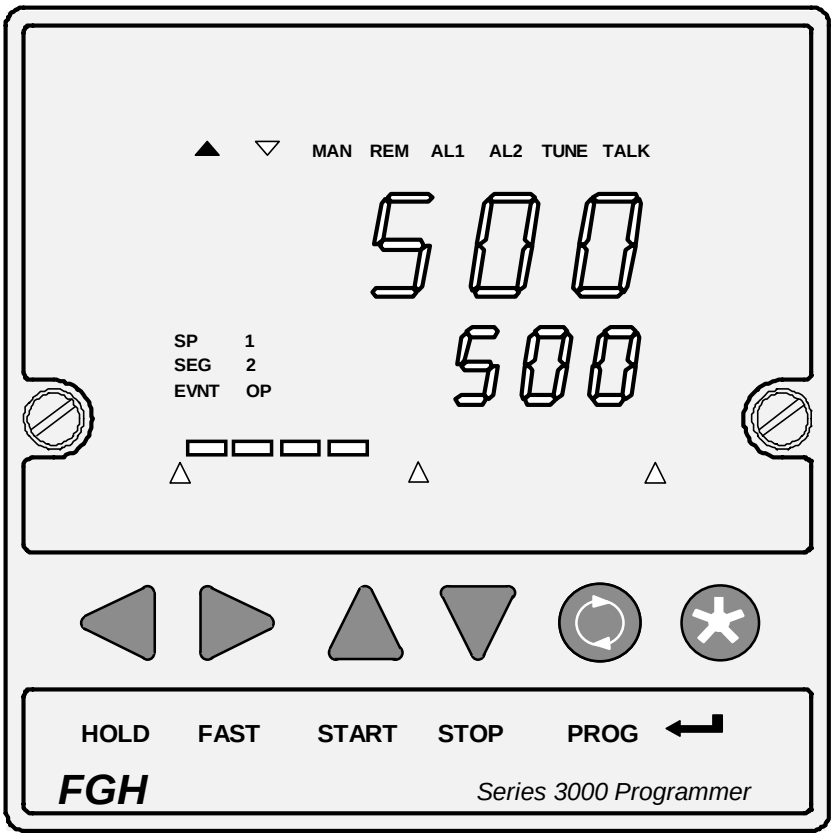
Communications Manual



Series 3000

FGH Controls Ltd
Openshaw Way,Letchworth
Herts. SG6 3ER
Tel: (01462) 686677
Fax: (01462) 480633

Contents.

1.0 GENERAL DESCRIPTION.

The FGH S3000 controller or P3000 programmer can be fitted with a serial communications option board. This board allows 2 way serial asynchronous communication with a computer.
Messages consist entirely of printable 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 supported.

1.1 Specification.

1.1.1 2-Wire EIA-485 Mode (RS485)

Transmission standard:	EIA-485 (RS-485)
Data rates:	1200, 2400, 4800 and 9600 baud.
Data format:	1 start, 7 data, odd parity, 1 stop bit.
Implementation:	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)

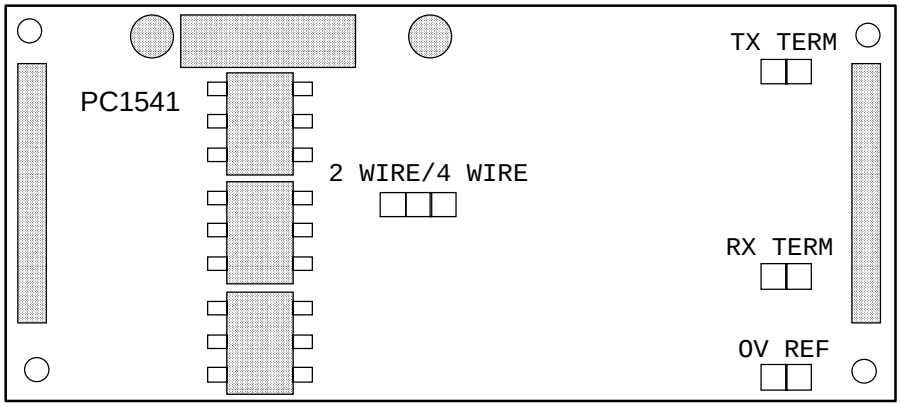
Transmission standard:	EIA-422-A (RS-422)
Data rates	1200, 2400, 4800 and 9600 baud.
Data format	1 start, 7 data, odd parity, 1 stop bit.
Implementation	4 wire full duplex.
Max drivers per line	32
Max receivers per line	32
Max cable length	1200 metres/3937 feet

1.2 Hardware Links

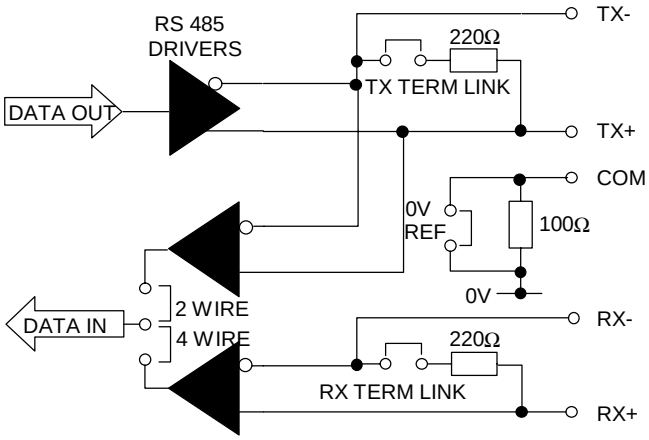
The following links are to be found on the comms board PC1541 inside the instrument and should be set as indicated.

PC1451 Link	EIA-485	EIA-422-A
TX TERM	Remove	Remove
RX TERM	Remove	Optional
4Wire/2Wire	2W	4W

Position of links on PC1541 Communications Board



The purpose of the various shorting links is described in the following paragraphs, 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.

2.4 PC-1541 Ground links

A 100 ohm resistor is fitted between COM and the instrument 0v. this can be modified by fitting link 0V REF which will then connect COM directly with the instrument 0v. 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 0V REF should be removed.

When the ***same*** Earth ***is*** connected to Frame Ground then link 0V 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.

2.0 CONNECTIONS.

2.1 General.

The series 3000 may be connected to any computer or device which supports the RS422 or RS485 interface standards. This includes any other communicating instruments such as the series 1000 or any other instrument using the FGH standard protocol.

The instrument uses a balanced voltage communications system which will perform well under most situations provided some simple guidelines are adhered to.

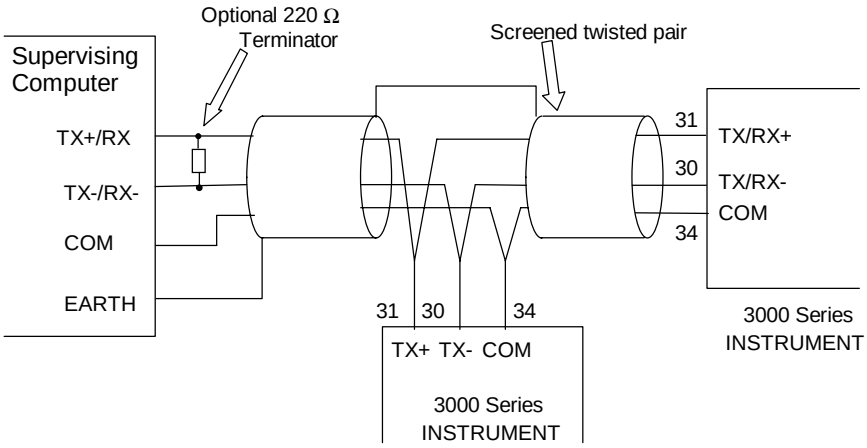
1. The communications wiring should be implemented using screened cable comprising one or two twisted pairs. The cable screen should be earthed at one point only.

2. The cable should be routed well away from sources of electrical noise such as motors, contactors and any other high voltage wiring.

3. The network should be wired as a daisy chain, taking the wires in to one instrument and hence on to the next. Wiring spurs should be avoided. Take care to continue the cable screen on to the next instrument.

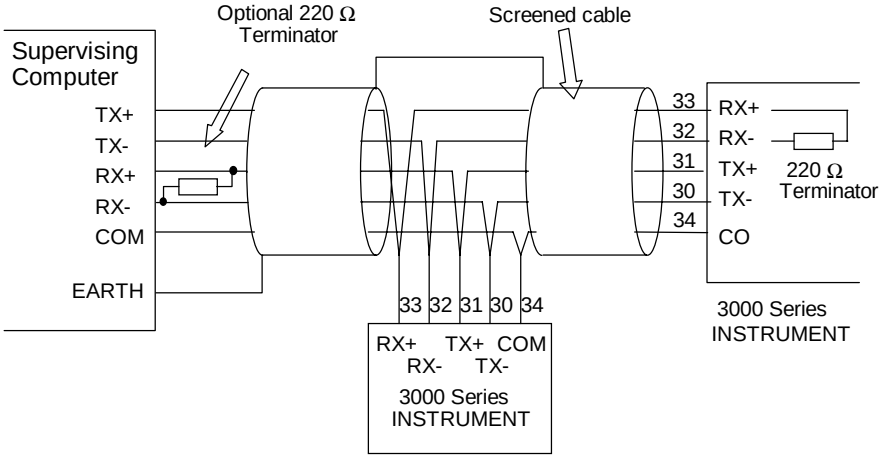
4. For long cable runs or noisy environments it may be necessary to fit a terminating resistor to the network. The terminator (a 220R resistor) should be fitted between RX+ and RX- on both the computer and the furthest instrument. For two wire networks this resistor should be fitted at the computer end only. Only one such resistor should be fitted on each wire pair. This termination resistor is fitted inside the instrument and may be selected by means of a jumper link

2.2 2 Wire EIA-485 Mode (RS-485)



This diagram shows the connection for EIA-485 mode. Note the use of screened cable. This connection method may be continued on to other instruments up to the maximum allowed.

2.3 4-Wire EIA-422-A (RS-422)



This diagram shows the connection for EIA-422 mode. Note the use of screened cable. This connection method may be continued on to other instruments up to the maximum allowed.

2.4 Connection reference table

Instrument Terminal	Function	EIA-485	EIA-422-A
30	TX-	TX-/RX-	TX-
31	TX+	TX+/RX+	TX+
32	RX-	N/C	RX-
33	RX+	N/C	RX+

34	COM	COM	COM
1 or 36	EARTH	N/C or SCREEN	N/C or SCREEN

2.5 Software Configuration.

Software configuration of the instrument may be performed once the instrument has been placed in 'Engineers mode'. This is done by removing the instrument from its sleeve and fitting the 'E mode' jumper link. The instrument is then replaced in its sleeve and power re-applied. Use the scroll button (>) to scroll through elements until the **ConS** group is reached. Press the star button to access the group.

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 P3000 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.

3.0 COMMUNICATIONS PROTOCOL

3.1 Message construction

Messages to and from the instrument vary in form depending on, amongst other things, the type of message and its contents. There are three basic message types:-

3.1.1 Write messages

Write messages to the instrument take the following form.

WAAPSSD..D<CR>

where **W** = write message header
AA = instrument address
P = parameter code
SS = secondary parameter field
D..D = numeric data field

messages to the instrument may have the fields separated by spaces. These spaces will be ignored. Messages from the instrument will not contain spaces. Eg.

W 45 C 0123 <CR>

will attempt to write 123 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.

3.1.2 Read messages

Read messages to the instrument take the following form.

RAAPSS<CR>

where **R** = read message header
AA = instrument address
P = parameter code to be read
SS = secondary parameter field

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

3.1.3 Set instrument status messages

Set messages to the instrument take the following form.

SAAC<CR>

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.

3.2 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 instrument. When this header is used the data field is absent. The **W** header is used to write data to the instrument. The **S** header is used to set the status of the instrument.

3.3 Instrument 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 instrument ignores the message unless it corresponds to its own address.

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.

3.3.1 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 both of the address characters. eg.

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.

3.4 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 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 3.7.

3.5 Secondary parameter field .

This field is used to specify the secondary reference required for some messages to the instrument. This should be a two digit field, left padded with a zero if required.

3.6 Message data field

The message data field consists of between four and eight ASCII characters and carries the data associated with the specified parameter. There are six types of data field, the type depending on the particular parameter being accessed. The parameter code tables in section 3.7 and 3.8 specify the data field format for each parameter.

3.7 Controller read/write parameter codes

Each parameter within the controller 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.
Please note that some parameters use coded data fields, the code meanings are listed after the table.

Code	SS Field	R/W	Parameter	Scroll Element	Data field type
@		R/W	Comms remote setpoint	-	1
A	00	R	Measured variable 1	-	1
	01	R	Measured variable 2	-	1
B		R/W	Output in 0.1% Desired valve position	OP OP	1 1
C	00	R/W	Local setpoint 1	SPLOC	1
	01 up	R/W	Terms set trigger setpoint	SP X	1
D	00	R/W	Propband in 0.1% (default)	ProP	1
	01 up	R/W	Terms set prop band in 0.1%	Pb X	1

E	00	R/W	Integral action time in S (default)	IAt	1
	01 up	R/W	Terms set integral tine in S	Iat X	1
F	00	R/W	Derivative action time in S (default)	dAt	1
	01 up	R/W	Terms set derivative time in S	dAt X	1
G		R/W	Derivative approach band in Xp	APr	1
H		R/W	Heat high power limit in 0.1%	H-PL	1
I		R/W	Heat TP cycle time in S	H CyC	1
J	00	R/W	Alarm level 1	ALr1	1
	01	R/W	Alarm level 2	ALr2	1
K	00	R/W	Alarm type code 1	A1tyP	1
	01	R/W	Alarm type code 2	A2tyP	1
L		R	Controller status	-	2
M	00	R/W	User Retransmit value 1	-	1
	01	R/W	User Retransmit value 2	-	1
N		R	Resultant (control) setpoint	-	1
O		R/W	Setpoint type code	StyP	1
P	00	R/W	Thermal head ratio in 0.1	rAtIO	1
	01	R/W	Ratio band in digits	bAnd	1
	02	R/W	Thermal head hi limit	th-hi	1
	03	R/W	Thermal head lo limit	th-lo	1
	04	R/W	Thermal head limit reference	rEF	1
	05	R/W	Max air setpoint	H-oP	1
Q		R	Instrument type code	-	3
R		R	Analogue remote setpoint value Slidewire position	-	1
S		R/W	Remote setpoint gain in 100ths	GAIN	1
T		R/W	Heat low power limit in 0.1% Cool high power limit in 0.1%	L-PL C-PL	1 1
U		R/W	Setpoint ramp rate in digits/hour	rAtE	1
V		R/W	Cool TP cycle time in S Valve action time in S	C CyC VAt	1 1
W		R/W	Cool relative prop band in 0.1	rEL	1
X		R/W	Heat/Cool deadband in 0.1% Motor valve deadband in 0.1%	dbAd dbAd	1 1
Y	00	R/W	Auxiliary setpoint 1	ASP1	1
	01	R/W	Auxiliary setpoint 2	ASP2	1
Z	00	R/W	Auxiliary output 1	AOP1	1
	01	R/W	Auxiliary output 2	AOP2	1

3.8 Programmer read/write parameter codes

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

Code	SS Field	R/W	Parameter	Scroll Element	DATA FIELD

B	N/A	R	Profile setpoint channel 2	-	1
C	N/A	R	Profile setpoint channel 1	-	1
D	N/A	R/W	Delay start time in mins	dELAy	1
E	N/A	R	Segment Elapsed time in mins	-	1
F	N/A	R/W	Channel 2 local setpoint	SPL 2	1
H	00	R/W	Default Profile hold band in digits	HoLd	1
	01 up	R/W	Terms set hold band	HB X	1
I	00	R/W	Default Profile hold type	HtyPE	1 (coded)
	01 up	R/W	Terms set hold type	Ht X	1 (coded)
J	N/A	R/W	Profile repeats	rEPtS	1
K	N/A	R	Number of repeats remaining	LEFt	1
L	YES	R/W	Channel 1 Seg target level	Li1XX	1
M	N/A	R	Current event status	Evnt	4
N	N/A	R/W	Ready mode event status	Er-r	4
O	YES	R	Channel 2 Seg target level	Li2XX	1
P	N/A	R/W	Profile pointer	-	1
Q	N/A	R	Profile status	-	5
R	YES	R/W	Segment event outputs	Er-XX	4
S	YES	R/W	Segment terms set number	Set X	1
T	YES	R/W	Channel 1 Seg target time	ti1XX	6
U	YES	R/W	Channel 2 Seg target time	ti2XX	6
X	N/A	R	Profile currently running	-	1

3.9 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. In the examples given the instrument address is assumed to be 20.

CODE	MESSAGE	REPLY	MEANING
M	S20M<CR>	*20M<CR>	Set controller to Manual mode
A	S20A<CR>	*20A<CR>	Set controller to Auto mode
P	S20P<CR>	*20P<CR>	Turn on the pretuner
O	S20O<CR>	*20O<CR>	Turn off pretuner
U	S20U<CR>	*20U<CR>	Unlatch any latched alarms

3.10 Programmer Set status codes.

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	S36S<CR>	*36S<CR>	Start profile pointed to by profile pointer
R	S36R<CR>	*36R<CR>	Reset currently running profile
H	S36H<CR>	*36H<CR>	Holds (pauses) execution of profile
F	S36F<CR>	*36F<CR>	Frees hold allowing profile to continue

3.11 Response from read or write

The response of the instrument to a satisfactory read or write message with the correct address will be as follows (unless an address wildcard is used, see para 3.3.1):

***AAPSSD..D<CR>**

The instrument will respond with a string of ASCII characters. The header will consist of '*' (Hex 2A).

The **AA**, **P** and **SS** fields will be exactly the same as in the message which prompted the response. The **SS** field will only be present for some messages to and from the instrument.

The format of the data field **D..D** is specific to the parameter code used and these field types are described in section 3.14.

3.12 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<CR>**

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.

3.13 Error responses

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

3.13.1 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<CR>

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

3.13.2 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<CR>

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

3.14 Data fields

There are six different type of data fields used on the Series 3000.

3.14.1 Data field type 1

This is the most common of the data field types and is used for simple numeric data. The data field consists of 4 numeric characters preceded by an optional minus sign.

For example to write to the local setpoint on instrument address 3.

The computer would send **W03C-0100<CR>**

The instrument would respond ***03C-0100<CR>**

Note that the data field is padded out to 4 characters in both the query and response messages.

3.14.2 Data field type 2

This data field type is reserved for the controller status obtained by reading parameter L. The instrument will respond with:

***NNLABCD<CR>**

Where **NN** is the instrument address.

A = Digital inputs = 0 Both off
= 1 Input 1 on, 2 off
= 2 Input 2 on, 1 off
= 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 tuner is off
= 1 tuner is on

D = Auto/Man = 0 Automatic
= 1 Manual

3.14.3 Data field type 3

This data type is reserved for the instrument type code. The instrument type code may be read from the controller by reading parameter code **Q**. The controller will respond with;

***NNQABCD<CR>**

Where **A** = Input 2 = 0 Controller with remote setpoint
= 1 Controller without remote setpoint
= 3 Programmer/Controller

BC = Input type
= 00 Type S, °C
= 01 Type R, °C
= 02 Type J, °C
= 03 Type K, °C
= 04 Type T, °C
= 05 Type E, °C
= 06 Type B, °C
= 07 Type N, °C
= 08 Type W, °C
= 09 Type W3, °C
= 10 Type W5, °C
= 11 Type NM, °C
= 12 Type L, °C
= 13 Type K10, °C
= 14 Type T10, °C
= 15 Type RT10, °C
= 16 Type RT, °C
= 17 Type S, °F
BC = Input type
= 18 Type R, °F
= 19 Type J, °F
= 20 Type K, °F

= 21 Type T, °F
= 22 Type E, °F
= 23 Type B, °F
= 24 Type N, °F
= 25 Type W, °F
= 26 Type W3, °F
= 27 Type W5, °F
= 28 Type NM, °F
= 29 Type L, °F
= 30 Type K10, °F
= 31 Type T10, °F
= 32 Type RT10, °F
= 33 Type RT, °F
= 34 Type Linear
= 35 Type Root

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

3.14.4 Data field type 4 (P3000 Only)

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.

3.14.5 Data field type 5 (P3000 Only)

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.

3.14.6 Data field type 6 (P3000 Only)

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 ***20T12E0000<CR>**
Indicating that the segment is programmed as an END.

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

3.15 Coded data fields.

3.15.1 Alarm type codes.

Reading and writing the controller alarm types is possible by use of parameters **K00** (alarm 1) and **K01** (alarm 2) where the data field of the message is coded as follows.

DATA FIELD	S3000 ALARM TYPE	P3000 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	REMOTE SP ACK' RELAY
0007	INVALID	PROGRAM RELAY
0008	INVALID	READY RELAY
0009	INVALID	UP RAMP RELAY
0010	INVALID	DOWN RAMP RELAY
0011	INVALID	SOAK RELAY

3.15.2 Setpoint type codes.

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

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

3.15.3 Ratio limit reference type codes.

The ratio limit reference (parameter code **P04**) is coded as follows.

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

3.15.4 Hold type codes (P3000 Only)

Programmer 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

APPENDIX A - BASIC COMMUNICATIONS PROGRAM

```
1000 REM DEMO PROGRAM FOR S1000 COMMUNICATIONS (GW-BASIC V3.23)
1010 ON ERROR GOTO 6000
1020 GOSUB 2000 'INITIALISE
1030 GOSUB 3000 'DRAW SCREEN
1040 WHILE PAUSED=0 'MAIN LOOP
1050   KEY (7) STOP:KEY (8) STOP
1060   FOR I=1 TO MAXPRM
1070     KEY (5) STOP:KEY (6) STOP
1080     LOCATE 5+I,1
1090     GOSUB 4000 'Send read question and get answer
1100     GOSUB 5000 'Display answer
1110     KEY (5) ON:KEY (6) ON
1120   NEXT I
1130   KEY (7) ON:KEY (8) ON
1140 WEND
1150 GOTO 1040
2000 REM *****  INITIALISE  *************
2010 ON KEY (5) GOSUB 7000 : KEY (5) ON 'Write
2020 ON KEY (6) GOSUB 10000 : KEY (6) ON 'Address
2030 ON KEY (7) GOSUB 11000 : KEY (7) ON 'Pause
2040 ON KEY (8) GOSUB 12000 : KEY (8) ON 'Cont
2050 ON KEY (10) GOSUB 9000 : KEY (10) ON 'Break
2060 WHILE PRAM$(MAXPRM) <> "END"
2070   MAXPRM = MAXPRM+1
2080   READ PRAM$(MAXPRM)
2090   READ PRMCODE$(MAXPRM)
2100   READ PRMSSFIELD(MAXPRM)
2110   READ PRMMUL(MAXPRM)
2120   READ PRMUNITS$(MAXPRM)
2130 WEND
2140 MAXPRM = MAXPRM-1
2150 OPEN "COM2:9600,O,7,1,DS0,CS0,CD0" AS #1
2160 ADDRESS$ = "00" 'Initial controller address
2170 RETURN
2180 REM DATA TABLE. TEXT(18 MAX),PARAMETER LETTER,SS FIELD,1/UNIT
       VALUE,UNITS
2190 REM USE SS FIELD=-1 FOR NOT USED
2200 DATA "MEASURED VALUE","A",0,1,"DEG"
2210 DATA "OUTPUT","B",-1,10,"%"
2220 DATA "LOCAL SET POINT","C",0,1,"DEG"
2230 DATA "PROPBAND","D",0,10,"%"
2240 DATA "INT ACTION TIME","E",0,1,"SEC"
2250 DATA "DER ACTION TIME","F",0,1,"SEC"
2260 DATA "DER APPROACH BAND","G",-1,10,"XP"
2270 DATA "END","",-1,0,""
2280 RETURN
3000 REM ***** DRAW SCREEN *************
3010 SCREEN 0,0:WIDTH 80
3020 CLS
3030 KEY 5,"WRITE"
3040 KEY 6,"ADDR'S"
3050 KEY 7,"PAUSE"
3060 KEY 8,"CONT"
3070 KEY 9,""
3080 KEY 10,"QUIT"
3090 LOCATE 2,24
3100 PRINT "F:G:H  S3000  COMMS DEMO"
3110 LOCATE 4,1
3120 PRINT TAB(4)"Parameter"TAB(23)"Question"TAB(33)"Answer"TAB(52)"Result"
```

```
3130 RETURN
4000 REM ***** READ AND GET ANSWER *************
4010 PRINT "->";
4020 TX$="R"+ADDRESS$+PRMCODE$(I)
4030 IF PRMSSFIELD(I)<0 THEN 4050
4040 IF  PRMSSFIELD(I)>9 THEN TX$=TX$+STR$(PRMSSFIELD(I)) ELSE
      TX$=TX$+"0"+RIGHT$(STR$(PRMSSFIELD(I)),1)
4050 TX$=TX$
4060 PRINT #1,TX$;CHR$(13);
4070 GOSUB 8000 'GET RX$
4080 RETURN
5000 REM ***** DISPLAY QUESTION AND ANSWER *************
5010 PRINT TAB(4) PRAM$(I); 'Text parameter title
5020 PRINT TAB(23) TX$;TAB(30);" =  ";RX$ SPACE$(5);
5030 VALUE$=RIGHT$(RX$,5)
5040 IF (LEFT$(VALUE$,1))>"-" THEN VALUE$=RIGHT$(VALUE$,4)
5050 PRINT TAB(50)" = ";
5060 IF (RX$="TIMEOUT") OR (LEFT$(RX$,1)<>"*") THEN GOTO 5090
5070 PRINT VAL(VALUE$)/PRMMUL(I);PRMUNITS$(I);"     ";
5080 GOTO 5100
5090 PRINT " ERROR"SPACE$(10);
5100 LOCATE CSRLIN,1:PRINT "   ";
5110 RETURN
6000 REM ***** ERROR HANDLER *************
6010 GOSUB 13000
6020 PRINT "  Error Type " ERR "At line "ERL
6030 LOCATE CURY,CURX:RESUME NEXT
7000 REM ***** WRITE TO CONTROLLER ************
7010 GOSUB 13000
7020 PRINT "Write Parameter ? ";
7030 WPRAM$=INKEY$:IF LEN(WPRAM$)=0 THEN 7030
7040 PRINT WPRAM$;
7050 LINE INPUT "     SS FIELD (## OR ENTER FOR NONE) ";WSS$
7060 PRINT TAB(26);"New Data [-]NNNN ";:LINE INPUT WDAT$
7070 WTX$="W"+ADDRESS$+WPRAM$+WSS$+WDAT$
7080 PRINT #1,WTX$;CHR$(13);
7090 GOSUB 8000 'GET RESPONSE RX$
7100 PRINT TAB(26);WTX$;" = ";RX$;
7110 LOCATE CURY,CURX
7120 RETURN
8000 REM ***** GET RESPONSE RX$ *************
8010 RX$="":IN$=""
8020 TIMLIMIT=TIMER+.5  'Timeout limit
8030 TIM=TIMER:TIMOUT=0:RXEND=0
8040 WHILE (RXEND=0) AND (TIMOUT=0)
8050   WHILE NOT EOF(1)
8060     IN$=INPUT$(1,#1)
8070     IF IN$=CHR$(13) THEN RXEND=1 ELSE RX$=RX$+IN$
8080     TIM=0
8090   WEND
8100   IF TIM>TIMLIMIT THEN TIMOUT=1
8110 WEND
8120 IF TIMOUT <>0 THEN RX$="TIMEOUT"
8130 RETURN
9000 REM ***** BREAK - KEY 10 *************
9010 LOCATE 23,1:END
10000 REM ***** CHANGE ADDRESS *************
10010 GOSUB 13000 'Position Cursor
10020 LINE INPUT;"     New Address  NN ";ADDRESS$
10030 LOCATE CURY,CURX
10040 RETURN
11000 REM ***** PAUSE *************
11010 GOSUB 13000 'Position cursor
```

```
11020 PAUSED=1:PRINT " ** Paused ** "
11030 LOCATE CURY,CURX:RETURN
12000 REM ***** CONT *************
12010 GOSUB 13000 'Position cursor
12020 PAUSED=0:PRINT " ** Continue ** "
12030 LOCATE CURY,CURX:RETURN
13000 REM ***** TALK WITH OPERATOR *************
13010 CURX=POS(0):CURY=CSRLIN 'Current cursor posn
13020 LOCATE 22,2:PRINT STRING$(77,32):LOCATE 22,2
13030 RETURN
```