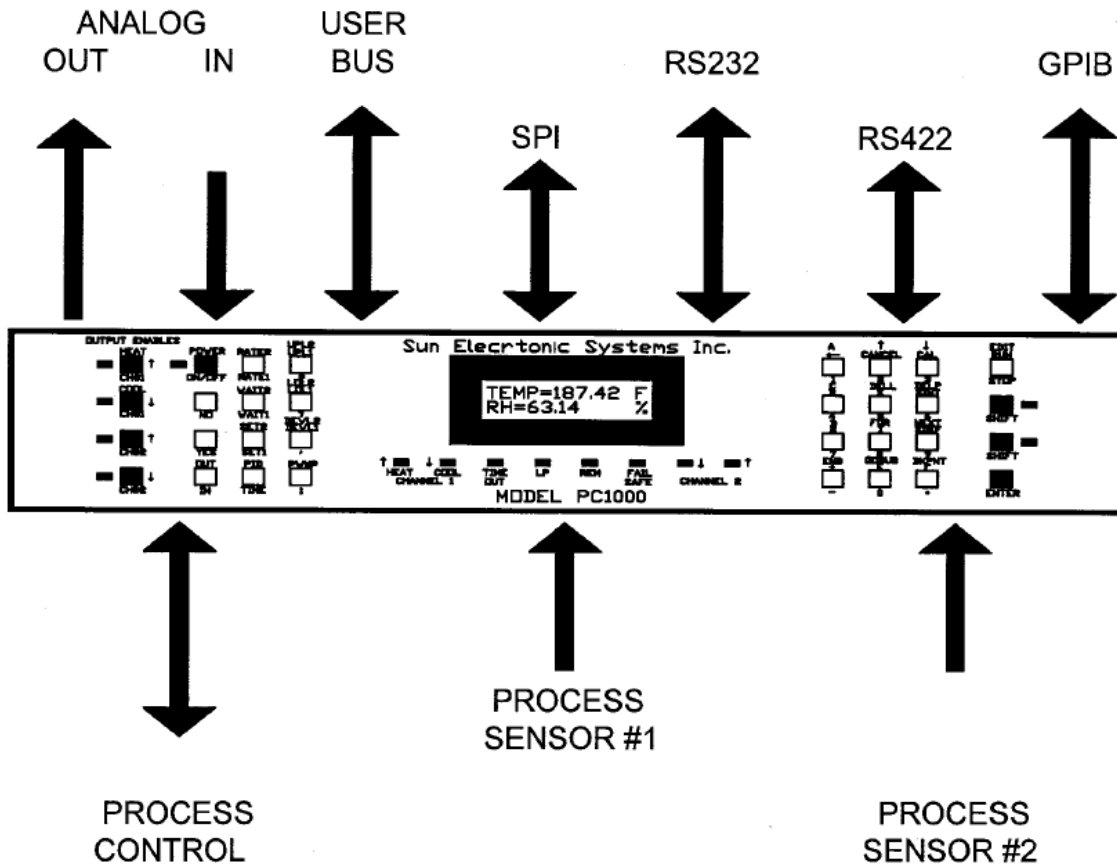


# MODEL PC1000

## PROCESS CONTROLLER

### USER MANUAL

REV: C  
0717



SUN ELECTRONIC SYSTEMS, INC  
1845 Shepard Dr.  
Titusville, FL 32780  
Tel: (321) 383-9400 Fax: (321) 383-9412  
Email: info@sunelectronics.com  
Website: www.sunelectronics.com

## **COPYRIGHT**

The information in this document has been carefully checked and is believed to be entirely reliable. No responsibility, however, is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the product described any license under the patent rights of SUN ELECTRONIC SYSTEMS, INC., or others.

Sun Systems strives constantly to improve the quality and performance of all of its products, in design, construction, and cost. As a consequence, SUN ELECTRONIC SYSTEMS, INC. reserves the right to change specifications, or other data contained herein, without notice.

Any software, firmware, designs, and any other programs in Sun Systems products remain the sole property of SUN ELECTRONIC SYSTEMS, INC. and may not be copied by any means. Purchase of said product constitutes agreement that software rights remain to the benefit of SUN ELECTRONIC SYSTEMS, INC.. Upon purchase, Sun Systems grants perpetual license for use by the customer.

## **LIMITED WARRANTY INFORMATION**

SUN ELECTRONIC SYSTEMS, INC. warrants this product to the original purchaser to be free from defects in material and workmanship and to operate within applicable specifications for a period of two (2) years from the date of shipment, provided it is used under normal operating conditions. This warranty does not apply to sealed assemblies which have been opened, or to any item which has been repaired or altered without SUN SYSTEMS authorization.

Risk of loss or damage to the product shall pass to the customer upon delivery, by SUN SYSTEMS, to the carrier at sun systems premises.

We will repair or, at our option, replace at no charge any of our products which in SUN SYSTEMS judgment, are found to be defective under the terms of this warranty. Except for such repair or replacement, SUN SYSTEMS will not be liable for any loss or damage to persons or property caused either directly or indirectly by use of this product or for any incidental damages or for any consequential damages. Before using, purchaser shall determine the suitability of this product via this document for his intended use and purchaser assumes all risk and liability whatsoever in connection therewith. No statement or recommendation not contained herein shall have any force or effect unless in agreement signed by an officer of SUN ELECTRONIC SYSTEMS, INC.

# Contents

<b>I INTRODUCTION .....</b>	<b>7</b>
FRONT PANEL OPERATION .....	8
REMOTE GPIB or RS232/422 OPERATION .....	8
PC1000 I/O INTERFACES .....	8
<b>II PC1000 INSTALLATION .....</b>	<b>9</b>
INITIAL INSPECTION AND UNPACKING .....	9
LOST OR DAMAGED EQUIPMENT .....	9
INPUT POWER VOLTAGE SELECTION .....	9
SENSOR INPUT CHANNELS .....	9
CALIBRATION MENU .....	10
CAL TO A STANDARD COMMANDS .....	11
PROCESS CONNECTION .....	12
EXAMPLE BIT ON/OFF CONTROL .....	12
PROCESS CONTROL VIA ANALOG OUTPUT .....	14
EXAMPLE ANALOG CONTROL .....	14
MOUNTING THE PC1000 .....	16
SETUP MENUS .....	16
SETUP DEFAULTS (SDEF MENU) .....	17
SET INTERRUPTS MENU (SINT) .....	18
PC1000 OPERATION EXAMPLES .....	19
FRONT PANEL SINGLE SEGMENT EXAMPLE .....	19
LOCAL PROGRAM EXAMPLE .....	20
<b>III PC1000 INPUT &amp; OUTPUT INTERFACES .....</b>	<b>23</b>
SENSOR INPUTS .....	23
SENSOR INFORMATION .....	23
ON/OFF PROCESS I/O PORT .....	24
ANALOG INPUT & OUTPUT PORT .....	27
RS232/422 REMOTE SERIAL INTERFACE .....	29
IEEE-488 (GPIB) INTERFACE .....	32
USER I/O PORT .....	34
HIGH SPEED SERIAL LINK INTERFACE (SPI) .....	37
<b>IV COMMAND OVERVIEW .....</b>	<b>39</b>
PROCESS CONTROLLING COMMANDS .....	39
PROCESS SAFETY LIMIT COMMANDS .....	40

CONTROL LOOP PARAMETER COMMANDS.....	41
LOCAL PROGRAM COMMANDS .....	41
POWER AND CONTROL LOOP ON/OFF COMMANDS.....	42
AUXILIARY I/O COMMANDS.....	43
GENERAL COMMANDS.....	44
Model TC01 Compatibility Commands .....	45
<b>V DETAILED COMMAND AND MENU INFORMATION.....</b>	<b>47</b>
BKPNT n.....	47
BKPNT Im.....	47
BKPNT?.....	48
C.....	48
C1? & C2? .....	49
CAL MENU .....	50
CAL TO A STANDARD .....	52
CHANNEL 1 & 2 OUTPUT ENABLES.....	53
C1ON+ C1OFF+ C1ON- C1OFF- C2ON+ C2OFF+ C2ON- C2OFF- .....	53
CSET1? & CSET2?.....	54
DELL.....	54
DELP .....	55
DEVL1 & DEVL2.....	56
EDIT.....	57
END .....	58
FOR/NEXT.....	58
GOSUB.....	59
I Variables .....	60
IN.....	61
LIST .....	62
LLO .....	62
LOL1 & LOL2.....	63
M.....	63
OFF.....	64
ON .....	64
OUT .....	65
PID Coefficient Modification .....	66
PIDA .....	67

PRINTn .....	67
PWMP .....	68
RATE1 & RATE2 .....	69
RTL .....	70
RUN .....	71
RUN - TIME OF DAY .....	72
S/I .....	73
SCALE1 and SCALE2 .....	74
SET1 and SET2 .....	75
SDEF MENU .....	76
SDEF .....	77
SINT MENU .....	78
SINT .....	79
STATUS? .....	80
STOP .....	81
STOPE9 .....	81
STORE .....	82
T .....	82
TIME .....	83
TIMEE? .....	83
UPPER LIMIT .....	84
UPL1 & UPL2 .....	84
VER? .....	84
WAIT1 & WAIT2 .....	85
#TEXT .....	86
? .....	86
<b>VI PID COEFFICIENT TUNING AND ADVANCED PID MODE CONTROL .....</b>	<b>87</b>
CONTROL SYSTEM RESPONSE .....	88
ITERATIVE PID COEFFICIENT DETERMINATION: .....	90
START WITH THE HEATING PID COEFFICIENT SET ( PID1+ ) .....	90
<b>VII PC1000 RELATIVE HUMIDITY CALCULATIONS .....</b>	<b>97</b>
<b>VIII PRINTER OPTION .....</b>	<b>99</b>
<b>IX FIRMWARE UPGRADE INFORMATION .....</b>	<b>101</b>
<b>X PC1000 LOCAL PROGRAM FUNCTIONAL FLOW CHART .....</b>	<b>103</b>
<b>XI PC1000 SPECIFICATIONS .....</b>	<b>105</b>

POWER REQUIREMENTS .....	105
MECHANICAL.....	105
ENVIRONMENTAL .....	105
SAFETY .....	105
PROCESS SENSOR INPUT CHANNELS.....	105
ABSOLUTE ERROR OVER RANGE (Note 1).....	105
OTHER FEATURES.....	106
IEEE-488 BUS INTERFACE (GPIB).....	106
RS232/422 SERIAL INTERFACE.....	106
4 CHANNEL, 8 BIT A/D INPUTS.....	106
4 CHANNEL, 8 BIT D/A OUTPUTS.....	106
DIGITAL PARALLEL USER PORT.....	106
HIGH SPEED SYNCHRONOUS SERIAL PORT .....	107
<b>XII PC1000 ERROR MESSAGES.....</b>	<b>109</b>
SRQ OVERFLOW .....	109

## I INTRODUCTION

Thank you for selecting the PC1000 process controller. Sun Electronic Systems Inc., a leading manufacturer of digital controllers and environmental test equipment, is happy to offer the PC1000 process controller. The PC1000 is a member of our family of microprocessor based digital controllers. It combines high accuracy with unparalleled functionality and performance. This combination addresses the needs of critical processes and complex control applications typical in the high performance arena.

As with all Sun Systems products, we would appreciate any comments, suggestions or criticisms that you may have or develop regarding your evaluation of this equipment. Please address your comments to:

Sun Electronic Systems, Inc.  
1845 Shepard Drive  
Titusville, FL 32780  
Tel: (321) 383-9400  
Fax: (321) 383-9412  
Email: sales@sunelectronics.com  
Website: www.sunelectronics.com

The MODEL PC1000 is a state of the art single/dual loop process controller with a wealth of additional control and monitoring abilities. To accurately control the process, the controller uses Proportional, Integral and Derivative (PID) control techniques to control one or both channels. The control loop output is in the form of time pulse width modulation for on off devices, or analog voltage or current output for continuous control when required. If necessary, you may tailor the characteristics of the PID algorithms independently for both channels. Further each channel has two sets of coefficients and the coefficients may be changed at any time, even during local program execution. The PC1000 is instructed to control the process variables, or I/O events, by issuing single control commands or by executing local control programs that are stored in its battery backed memory. Single commands and local programs may be issued from either the local control panel or remotely from either the RS232/422 serial interface or the GPIB interface.

Process control usually involves controlling a process variable at one or at several different set points. In the case of more than one set point, the rate of change from one set point to another and the time waiting at a particular point are considerations. The PC1000 accepts commands defining an individual set segment, containing rate of change from the present process value to the final set point, wait time at the final set point, and final set point information. At the end of a wait period, notification is given to the source of the command. When controlling the PC1000 via a remote bus, the host computer is free to perform other tasks since the PC1000 automatically performs the ramping and timing functions. If multiple segments are to be performed by the PC1000 automatically, it can be programmed by entering an EDIT mode and the local program is executed in the RUN mode. The PC1000 stores up to 10 local programs into battery backed memory. The number of commands resident in a local program is limited to approximately 100 by memory space. However since programs can call other programs as subroutines and since FOR-NEXT looping is provided very complex profiles can be programmed. Single, local program and remote commands are also provided for things other than controlling the process variables, such as input and output to A/D's, D/A's, user bus, etc. The PC1000 was designed to be as versatile and user friendly as possible.

To configure the PC1000 for your application, three local menus are provided. The CAL menu permits selection of probe/sensor type, calibration and scale selection for each channel independently. The SDEF menu permits modification of the PC1000 default parameters such as GPIB bus address, serial bus protocol, D/A configurations, etc. Lastly the SINT menu permits modification of interrupt configuration parameters such as audible alarm functions, GPIB service request functions, etc.

## FRONT PANEL OPERATION

The PC1000 can operate in a standalone configuration through the use of its control panel located on the front of the unit. The 32 key keyboard contains keys for parameters, commands, program instructions, program editing and menu selections. The 32 character alphanumeric display provides process status and operation prompts in easy to understand English. Every effort has been made to provide a user friendly environment.

To permit easy command entry and local program generation, the following operators on the local control panel permit command entry, command cancellation, menu decisions, single command line editing and program editing during EDIT mode.

UP,DOWN ARROWS	;Line movement during Local Program editing.
DELL	;Delete current local program line
DELP n	;Delete local program number n
LEFT ARROW	;Character backspace during command entry.
CANCEL	;Cancels present command entry.
ENTER	;Enters the command line.
	;Or inserts a line into the local program.
YES, NO	;Operators for SDEF, SINT and CAL menu's.

### Parameter Entry:

Parameters are used for numeric data entry, delimiters and function operators in command lines.

0-9 .+ -	Numeric data (depicted as nnn.n in following sections)
:	Numeric delimiter.
e or E	Exponent for scientific notation.
=	Set variable left side equal to value right side.
?	Examine variable left side.

ALSO: spaces are ignored and upper/lower case is equivalent

## REMOTE GPIB or RS232/422 OPERATION

The PC1000 can operate remotely from a host computer via an IEEE-488 standard GPIB interface bus or over an RS232 or RS422 serial interface. Individual commands may be sent to the PC1000 or whole programs may be downloaded and executed, further relieving the host computer. Local timing and a rich selection of interrupts and alarms insure that synchronization between the PC1000 and host computer is maintained. Section IV contains a list of commands and program instructions for the PC1000.

## PC1000 I/O INTERFACES

In addition to the 2 process sensor inputs the PC1000 provides 4 analog input and 4 analog output channels, auxiliary relay control, auxiliary status inputs, a user digital parallel I/O port and a high speed serial I/O link. Configuration and use of the auxiliary ports is controlled by local menu's and commands that provide read and write capability to each interface. For example, the SDEF menu allows you to configure two of the analog output channels for automatic chart recorder output of the process variables. Section III provides detailed information on the I/O ports and the commands to access them are covered in section IV.



## II PC1000 INSTALLATION

### INITIAL INSPECTION AND UNPACKING

Inspect the shipping container for obvious damage. If the shipping container is damaged, then a written note on the bill of landing describing the damage should be made while the delivery person is still on the premises. Unpack the PC1000 and save all carton and cushioning material in case the unit needs to be returned. Verify the contents of the carton matches that of the items listed on the packing slip.

### LOST OR DAMAGED EQUIPMENT

If you determine that the goods described on the packing slip are lost or damaged, first notify the transportation company to obtain the correct procedure for returning merchandise for repair or replacement. Then call Sun Systems for a return authorization number, so that we may best handle the merchandise when it is returned.



### INPUT POWER VOLTAGE SELECTION

Selection of 110 VAC or 220-240 VAC is accomplished by rotating the fuse holder in the power input connector on the back panel. The silk-screen arrow on the back panel must point to the correct voltage to be used. Do not plug the PC1000 into the line as yet.

**Note:** If your PC1000 was purchased as part of a system that was ordered from a vendor other than SUN, the sensor and process connections probably have already been done. If so then skip ahead to MOUNTING THE PC1000.

### SENSOR INPUT CHANNELS

The two process sensor input channels are each 15 bit precision electrically isolated and software configured. Stable analog circuitry and high conversion resolution insures adequate precision for even the most demanding applications. Electrical isolation permits the process sensors to be attached to devices containing voltages up to 240 volts differential from each other or the

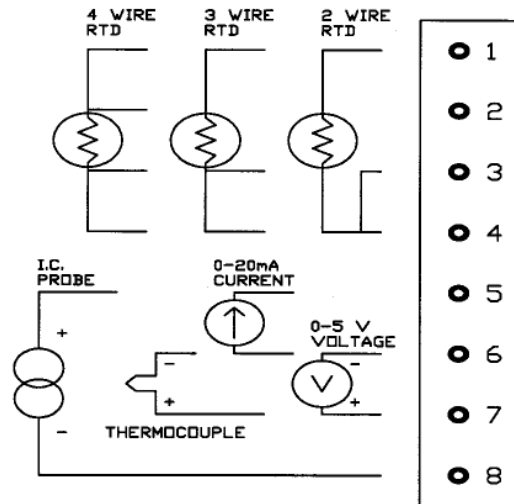
PC1000 circuitry. Via the CAL menu, each channel is independently configured for RTD temperature probe, J, K or T thermocouple, Integrated circuit temperature probe, Voltage input or Current input. Except for the temperature/humidity mode, each channels linearized measured value is displayed on the front panels 2 line 32 character alphanumeric display, and of course may be read remotely. When in dry-wet bulb temperature mode or dry-dew mode, dry bulb temperature and calculated relative humidity are available.

To measure your process, your sensors must now be connected to the PC1000 sensor input connectors. The following figure provides sensor connection information:

Now that the sensors are connected, you need to tell the PC1000 what kind of sensors you are using and to calibrate them. To do this plug in the PC1000 and turn it on. The CAL menu permits you to select a sensor type, calibrate the sensor, and/or to select the units of measure for the sensor. To calibrate a sensor using this menu, it is necessary to place the sensor at requested cal points, such as 0 deg C and 100 deg C when a temperature sensor is selected. Zero and one hundred were chosen since it is easy to create them using ice water and boiling water. It is not necessary to provide great accuracy in the CAL menu if the cal to a standard is then used for fine calibration.

When a new type of probe is to be selected this cal routine needs to be run. If you are not changing the sensor type but just recalibrating then the "cal to a standard" commands may be used. The CAL menu will show the current probe types. This menu operates only from the local front panel. The following is a sample of the cal menu dialog.

PROBE/SENSOR CONNECTIONS  
CHANNEL #1 or #2



## CALIBRATION MENU

green CAL then ENTER to enter calibrate menu

**Note:** after CAL is entered the PC1000 will display the current types of probes that it is currently configured for. The top LCD line will show probe type for channel 1 while the bottom line will give probe type for channel 2. The display will last for about 3 seconds. Then;

CAL SENSORS? (Y/N) enter yes to calibrate chan 1 or 2  
enter no to only change scale(s)

ENTER CAL ACCESS CODE

At this point the menu is asking for an access number to lessen the chance of unauthorized or accidental entry into the cal routine. The default code is 9 and may be changed to another number in the SINT menu

CAL CHAN #1? (Y/N) enter yes to calibrate chan 1

If yes to cal chan 1 then select probe type as;

**Note:** once a type is selected you are committed to calibrating this sensor. Calibration will only be approximate if defaults are used.

RTD .385 PROBE [Yes/No]?	-200 to +320 C [ DIN 43760 ]
RTD .392 PROBE [Yes/No]?	-200 to +320 C
J NARROW [Yes/No]?	-200 to +320 C J THERMOCOUPLE
J WIDE [Yes/No]	-200 to +750 C
K NARROW [Yes/No]?	-200 to +320 C K THERMOCOUPLE
K WIDE [Yes/No]	-200 to +1250 C
T THERMOCOUPLE [Yes/No]?	-200 to +320 C T THERMOCOUPLE
SOLID STATE [Yes/No]?	-60 to +160 C IC TEMP SENSOR
0 TO 5 VOLT IN [Yes/No]?	0V min to 5V max range
0 TO 20 mA IN [Yes/No]?	0mA min to 20mA max range

If a temperature probe was selected;

PROBE AT 0 DEG C [Yes/No]?	Set probe to 0 deg C and enter Yes when ready. NO=use default.
PROBE AT 100 DEG C [Yes/No]?	Set probe to 100 deg C and enter Yes when ready. NO=use default.
DESIRED SCALE? 1=C,2=F,3=K	Enter number to select scale for the current channel.

If voltage or current was selected;

LOW SCALE INPUT [Yes/No]?	Set low voltage or current to input Yes when ready. NO 0 volt default.
LOW SCALE VALUE ?	Enter the number that represents the process.
HIGH SCALE INPUT [Yes/No]?	Set high voltage or current to input Yes when ready. NO for 5 volt default
HIGH SCALE VALUE ?	Enter the number that represents the process.

CAL CHAN 2 [Yes/No]? Repeat similar process as above channel 1.

If a no answer was given to "CAL PROBES" prompt, then you are asked if you wish to change the default units of measure for each channel.

## CAL TO A STANDARD COMMANDS

Cal to a standard allows for fine calibration to your in house standard instrument. Three commands are supplied to perform the calibration. "CALA" performs an offset cal to the first process point while "CALB" performs a gain cal between the first point and the second process value supplied with the CALB command. "CALC" clears both the offset and gain cal adjustments.

To perform the cal to a standard:

1. Clear existing standard cal if any  
CALC, access code
2. Place your standard probe as close as possible to the probe or probes to be calibrated.
3. Set the process to go to the first calibration point.
4. Allow time for process stabilization.

5. To cal the channel 1 probe offset;  
CALA1, access code, standard reading
6. To cal the channel 2 probe offset;  
CALA2, access code, standard reading

Where; access code is the CAL menu access code number and standard reading is the reading that your house standard instrument is reading.

7. Set the process to go to the second calibration point. The larger the difference between the two points the more accurate the gain cal will be.
8. Allow time for process stabilization.
9. To cal the channel 1 probe gain;  
CALB1, access code, standard reading
10. To cal the channel 2 probe gain;  
CALB2, access code, standard reading

## PROCESS CONNECTION

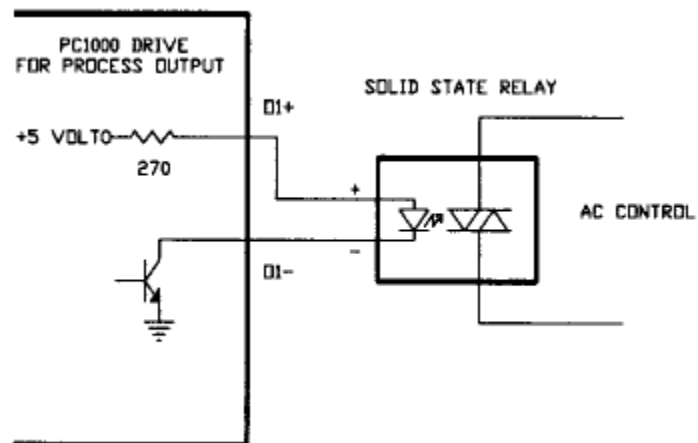
### EXAMPLE BIT ON/OFF CONTROL

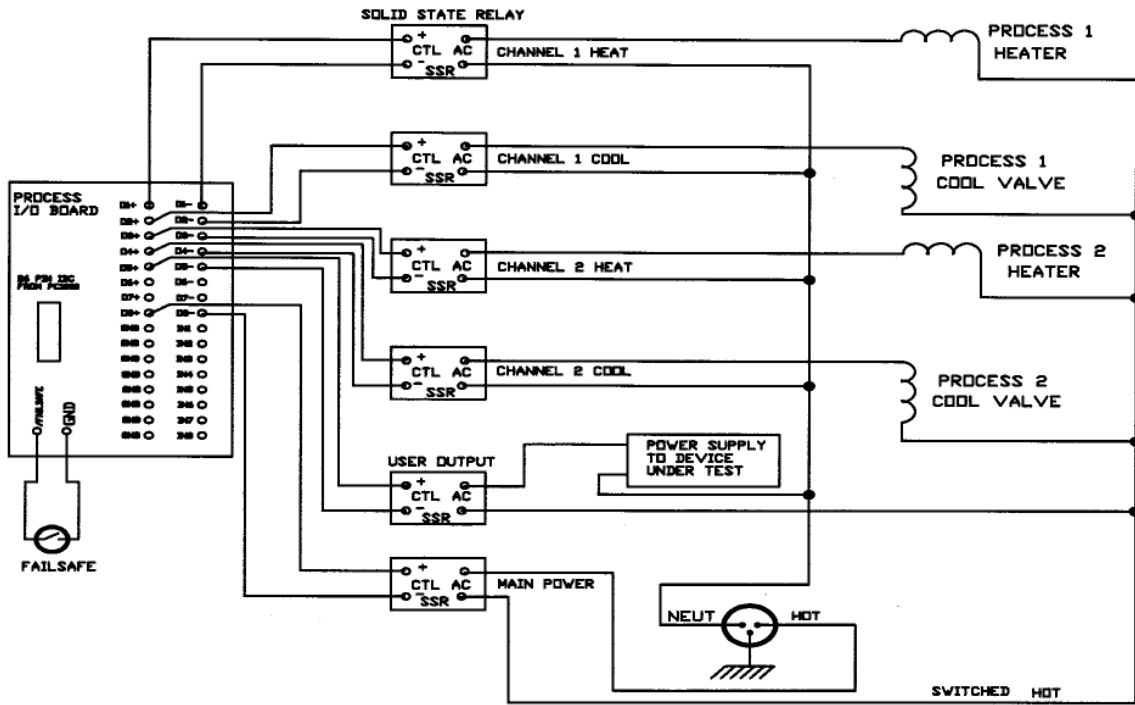
The PC1000 monitors the process with its probe inputs, calculates what it needs to do, then controls the process via on/off or optionally analog outputs. Section III provides detailed information on the PC1000 interfaces. In this section sufficient information is given for hookup.

Plug the flat cable into the PROCESS port on the rear of the PC1000. Plug the other end into the process I/O printed circuit board that is provided. The process I/O board provides screw terminals for connection to your equipment. The 26 pin flat cable provides low level signals to and from the process that you wish to control. Mount the I/O board in your equipment. The following provides information on the process I/O board connections.

- On+ ;output 270 ohm pull up to +5 volt (n=1 to 8)
- On- ;open collector output (n=1 to 8)
- INn ;input (n=1 to 8) ,has 10 K pullups to 5 volt
- GND ;PC1000 ground
- /FAILSAFE ;bring to GND to place PC1000 in failsafe, has 10 K pullup to 5 volt

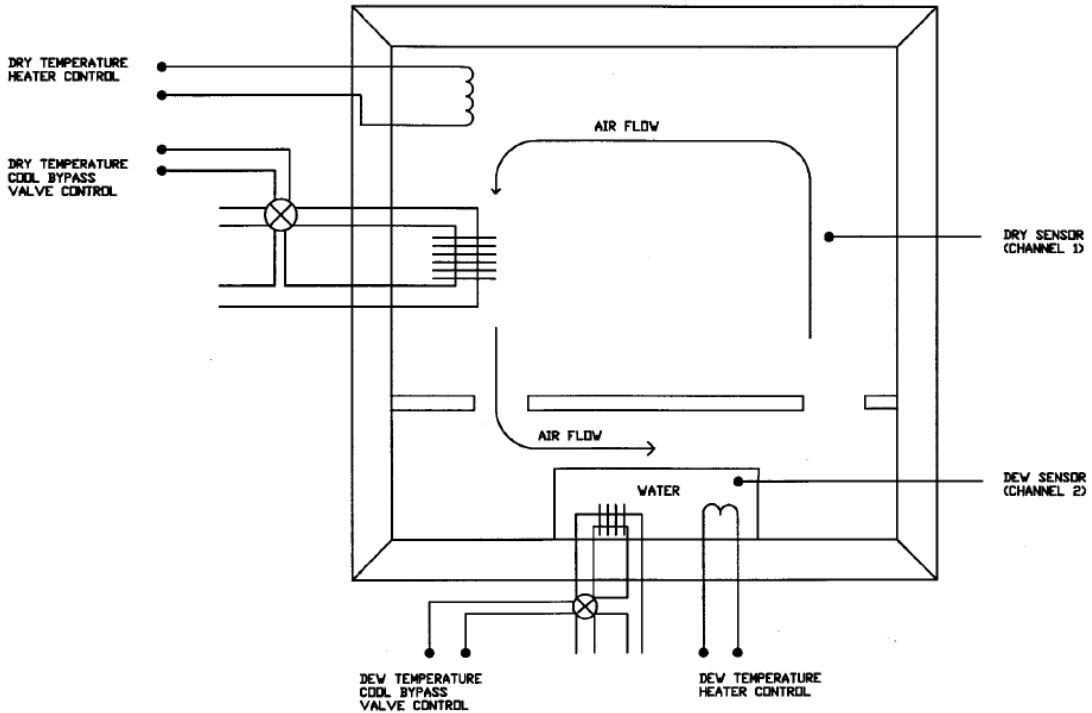
Next connect your solid state relays into your process circuits and to the process I/O board. Note that the PC1000 outputs are designed to drive solid state relays that switch the power to heaters, cool valves, etc. The PC1000 outputs are not able to drive power loads directly. Solid state relay modules can be purchased from SUN or through normal distribution. They are available with a number of voltage and current ratings. The PC1000 turns on the solid state relay (SSR) by sourcing current from the 5 volt pullup, through the SSR input light emitting diode, to ground via the open collector output. The following is an example connection to a dual channel temperature control process.

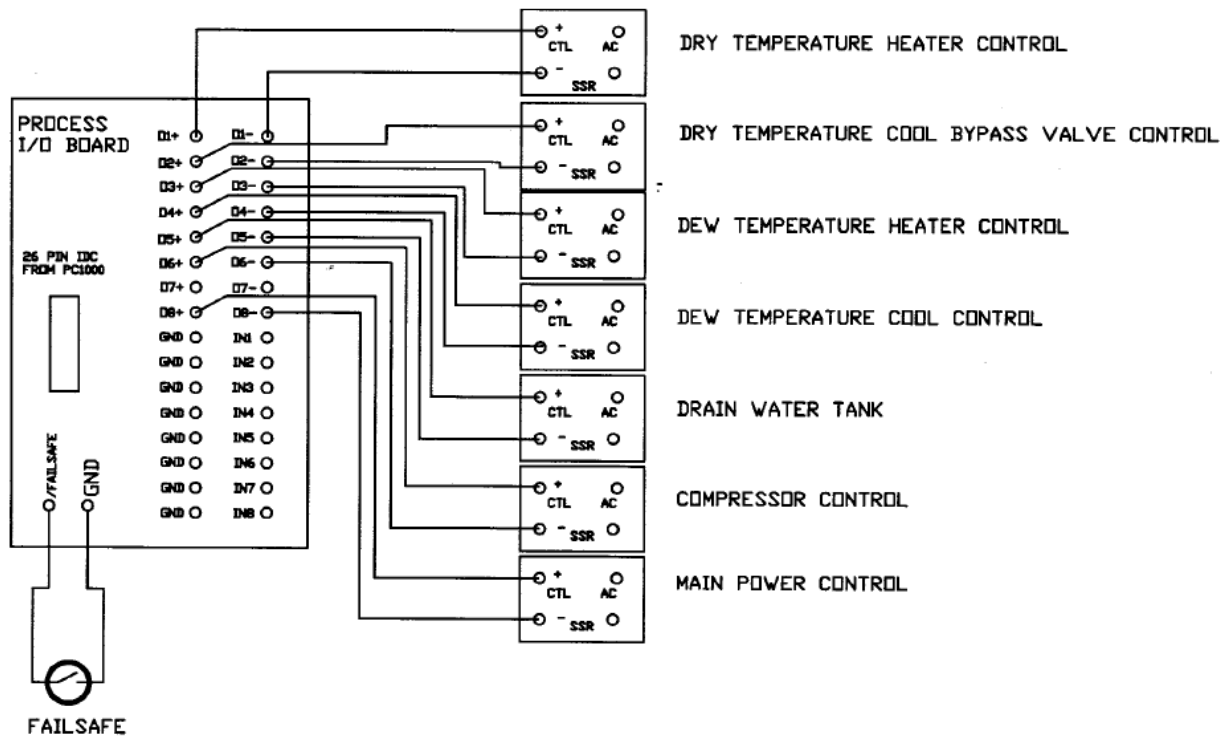




Also note that an external fail-safe input is provided for added safety.

The following example illustrates using the PC1000 in its dual channel control mode. This application deals with the control of temperature and humidity using the dry/dew point method. In the chamber, air passes over the bath of water. That air becomes saturated at the water temperature. The PC1000 will maintain the water temperature at the dew point temperature associated with the dry temperature and the desired relative humidity. The PC1000 automatically controls the dry and dew temperatures based on the values entered for temperature and percent relative humidity. Channel 1 measures and controls the dry temperature. Channel 2 measures and controls the water at the dew temperature that the PC1000 calculates.





## PROCESS CONTROL VIA ANALOG OUTPUT

### EXAMPLE ANALOG CONTROL

If your process requires an analog voltage, rather than on/off control, then the PID control loop signals may be obtained, in linear form, at the ANALOG rear panel port. The pin outs that are used are:

For channel #1	PIN #7=ANALOG OUT C
	PIN #10 & #11=GROUND
For channel #2	PIN #8=ANALOG OUT D
	PIN #10 & #11=GROUND

The PID to analog output is enabled in the SDEF menu. When you answer yes to the appropriate PID to D/A, you will be asked to enter a mode number 0 to 3. The mode determines how the PC1000 outputs the PID control signal, as follows;

MODE=0	low=full cool, mid=no change, high=full heat
MODE=1	low=no heat, mid=1/2 heat, high=full heat when cool is required, output=low
MODE=2	low=no cool, mid=1/2 cool, high=full cool when heat is required, output=low

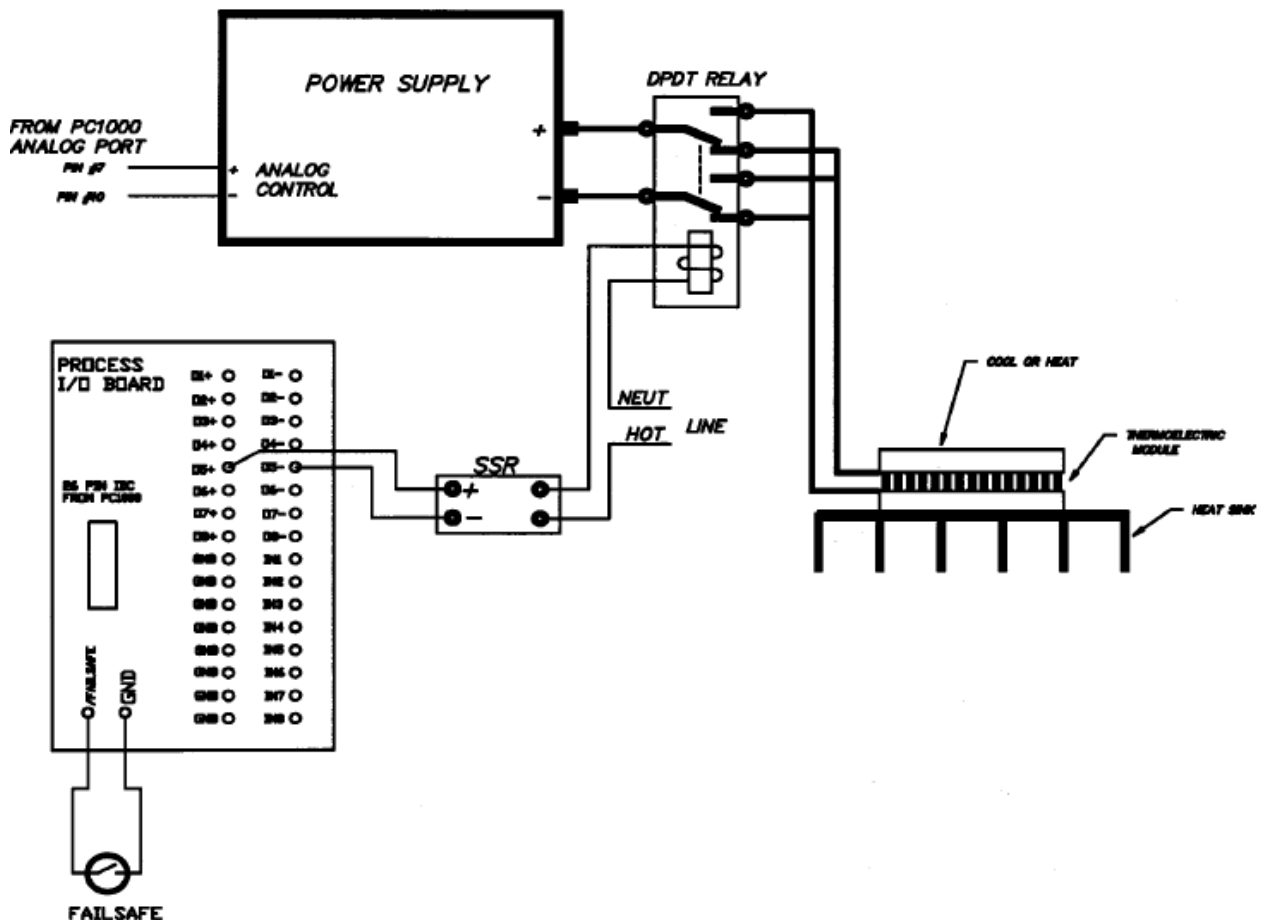
MODE=3

low=no heat/cool, mid=1/2 heat/cool, high=full process output bit #5 channel 1, or bit #6 channel 2, will be on when heat is required and it will be off when cool is needed.

Where: low is low output voltage  
mid is mid range voltage  
high is maximum output voltage

The voltage range for each analog output is selected as 0 to 5 volt, or -5 to +5 volt in the SDEF menu. For convenience we have used cool to represent decreasing the process and heat for increasing the process.

To illustrate the use of the analog process control outputs, assume that you need to control a linear power supply that will power a thermoelectric module. The power supply has an analog control input that requires 0 to 5 volts for off to full supply output. The module will cool if the current flows in one direction and it will heat if the current is reversed. If we connect the supply, thermoelectric module, and PC1000 in the following way;



Now in the SDEF menu we set D/A output C as 0 to 5 volt range, output PID #1 to D/A output C, and select MODE=3. The 0 to 5 volt analog output will control the power supply and process output bit #5 will automatically control the direction of current flow through the thermoelectric module.

## **MOUNTING THE PC1000**

Ears are supplied for 19" rack mounting. For table top use, a bail is provided on the bottom of the unit. The bail and/or the rack ears may be removed as needed. The length of the sensors is selected to suit your needs. A longer 26 conductor flat cable may also be used, to fit your installation requirements.

## **SETUP MENUS**

Now that the PC1000 is connected to your process, it is time to configure it to your requirements. This is performed by running two menus, the SDEF menu and the SINT menu, then setting the internal time of day.



## SETUP DEFAULTS (SDEF MENU)

The PC1000 contains default values which may be tailored to your individual system requirements. The values are stored in nonvolatile memory eliminating the need to reload after a power interruption. The default values may be set locally from the control panel by depressing the SDEF and ENTER keys followed by responding to the menu questions on the control panel display. If you are unsure of the settings that you will require then just make sure that the line frequency is set correctly. As you run the menus the current setting is shown within parentheses. To keep the current setting and go on to the next prompt, just press the enter key. For further information see the detailed command section.

SDEF then ENTER to run set defaults menu

```
SET DEFAULTS? (Y/N)          enter yes to set new defaults

    DUAL LOOP CTL? (Y/N) {NO if controlling 1 channel}
        AUTO RH CHANNEL 2 ?          {calculate rel humidity}
            WET DRY HUMIDITY? (Y/N)
    GPIB ADDRESS (0-30):
    GPIB LOCKUP TIME (2-59sec):
    TIMEOUT PRETIME (0-59sec):
    POWER DOWN AUTO RESTART (0-59min):
    D/A OUTPUT A (1=0 to 5,2=-5 to +5):
    D/A OUTPUT B (1=0 to 5,2=-5 to +5):
    D/A OUTPUT C (1=0 to 5,2=-5 to +5):
    D/A OUTPUT D (1=0 to 5,2=-5 to +5):
    SERIAL PORT MODE(1=232, 2=422):
    RS CHAR ECHO ON (Y/N):
    BAUD RATE (1=9600 to 6=300):
    RS DTR/CTS ON (Y/N):
    BUZZER VOLUME (0=OFF,to 3=LOUD):
    LINE FREQ (1=60HZ,2=50HZ):
    CHART CHAN #1 ON A (Y/N)          {analog charting}
        LOW CHART VALUE?
        HIGH CHART VALUE?
    CHART CHAN #2 ON B (Y/N)          {analog charting}
        LOW CHART VALUE?
        HIGH CHART VALUE?
    PID#1 TO C D/A? (Y/N)            {analog process ctl}
        MODE 0-3 ?
    PID#2 TO D D/A? (Y/N)            {analog process ctl}
        MODE 0-3 ?
*   ENABLE LINE PRINTER
*   LINE PER MINUTE 1 TO 6
*   LOW CHART VALUE?                {channel 1 charting}
*   HIGH CHART VALUE?
*   LOW CHART VALUE?                {channel 2 charting}
*   HIGH CHART VALUE?
    +/- CH #1 TRIGGER ?              {when to start wait1 counting}
    +/- CH #2 TRIGGER ?              {when to start wait2 counting}
```

\* = only if optional printer is installed

## SET INTERRUPTS MENU (SINT)

The PC1000 provides for local alarms and for interrupts to the communication ports. The SINT menu is provided so that you may configure these features. The settings are stored in nonvolatile memory eliminating the need to reload after power interruption. The interrupt values may be set locally from the front panel by depressing the SINT , ENTER keys then responding to the menu prompts on the front panel display. If you are unsure of the settings you need at this point then skip running the SINT menu for now. For further information see the detailed command section.

SINT            to enter set interrupts menu

SET INTERRUPTS? (Y/N)    ;YES to enter menu

DISABLE ALL INTERRUPTS (Y/N)	;YES to disable
BUZZER LOCAL PROGRAM TIMEOUT (Y/N)?	;YES for buzzer
BUZZER LOCAL PROGRAM DONE (Y/N)?	;YES for buzzer
BUZZER SINGLE TIMEOUT (Y/N)?	;YES for buzzer
INTERRUPT LOCAL PROGRAM TIMEOUT (Y/N)?	;YES for int.
INTERRUPT LOCAL PROGRAM DONE (Y/N)?	;YES for int.
INTERRUPT SINGLE TEMP TIMEOUT (Y/N)?	;YES for int.
DEVIATION INTERRUPT (Y/N)?	;YES for int.
COMMAND ERROR INTERRUPT (Y/N)?	;YES for int.
BKPNT INTERRUPT (Y/N)?	;YES for int.
P_POLL (1-8) 0=NO :	;number of bit

See SINT in the detailed command information section for information on the interrupts that the PC1000 generates. Also see section III for remote control information.

## SETTING TIME OF DAY CLOCK

The last thing that needs to be set is the internal time of day clock so that it is set to your local time. To set the time enter the following from the front panel;

TIME=hh:mm:ss            Where hh:mm:ss is in 24 hour format  
                              For example one thirty in the  
                              afternoon = 13:30:00

## PC1000 OPERATION EXAMPLES

### FRONT PANEL SINGLE SEGMENT EXAMPLE

After the PC1000 is set up the following provides an example of how to program a single mode segment. In this example temperature is the assumed process variable.

1. First set the channel 1 rate of change of temperature to 30 deg per minute.

```
RATE1
=
3
0
blue ENTER
```

2. Next set the channel 1 wait time to 1 minute and 10 sec

```
WAIT1
=
0
:
1
:
1
0
blue ENTER
```

3. Then set channel 1 target temperature to 40 degrees

```
SET1
=
4
0
blue ENTER
```

To enable the heat and cool outputs, press the yellow channel 1 heat on/off and cool on/off keys. The process will ramp to 40 deg and after 1 minute 10 sec will time out.

## LOCAL PROGRAM EXAMPLE

The following program demonstrates some of the basic local program features. This example assumes that the PC1000 is configured as a two channel temperature controller. The program will cause process variable #1 to go to 50 deg at a rate of 30 deg per minute and stay there for 5 minutes. After the 5 minutes at 50 deg the buzzer will buzz and BKPNT 1 will be displayed. To continue from the breakpoint press ENTER. Process variable #2 will then go to 0 deg, at a rate of 25 deg per minute and stay there for 7 and one half minutes. Then BKPNT 2 will be executed, press enter to continue. Then the above will be repeated 5 times by the FOR-NEXT loop. To enter the program in program #0 perform the following key strokes.

```
red shift                                ;to delete existing prog #0 (if any)
DELP
0
blue ENTER
ARE YOU SURE? [Y/N]                      ;response from PC1000
YES
blue ENTER

red shift                                ;to create a program
EDIT
0
blue ENTER                                free memory is displayed, then line #1
                                           is displayed , if program is empty then
                                           END is displayed
```

**Note:** While in edit mode red up and down arrow keys allow you to move in the local program. The red DELL allows you to delete the current displayed program line. A line can be inserted in front of the current displayed program line by entering the new line.

```
red shift
FOR
0
=
1
,
6
blue ENTER

RATE1
=
3
0
blue ENTER

WAIT1
=
5
blue ENTER

SET1
=
5
0
blue ENTER
```

red shift  
BKPNT  
1  
blue ENTER

green shift  
RATE2  
=  
2  
5  
blue ENTER

green shift  
WAIT2  
=  
0  
:  
7  
:  
3  
0  
blue ENTER

green shift  
SET2  
=  
0  
blue ENTER

red shift  
BKPNT  
2  
blue ENTER

red shift  
NEXT  
0  
blue ENTER

red shift  
END  
blue ENTER ; local edit now done

To run the program, green shift, RUN, 0, ENTER



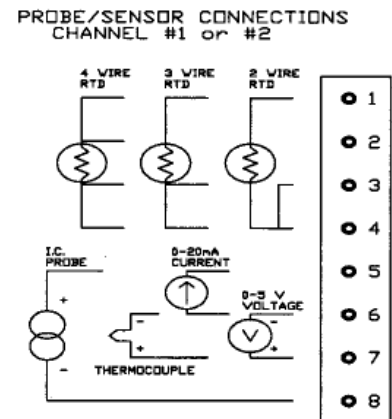
### III PC1000 INPUT & OUTPUT INTERFACES

#### SENSOR INPUTS

The PC1000 provides 2 electrically isolated process sensor input channels. The process sensor input channels accept several different types of sensors. The type of sensor, calibration of the sensor and units of measure is selected in the CAL menu, refer to section V. The following represents the connections to the removable probe connectors located on the PC1000 rear panel

Sensor input wiring (8 pos screw connector)

PROBE TYPE	CONFIGURATION INFORMATION
2 WIRE 100 ohm RTD	PIN 1=RTD OUT, PIN 3,4=RTD IN
3 WIRE 100 ohm RTD	PIN 1=RTD OUT, PIN 3,4=RTD IN
4 WIRE 100 ohm RTD	PIN 1,2=RTD OUT, PIN 3,4=RTD IN
THERMOCOUPLE	PIN 7=+LEAD, PIN 6=-LEAD
SOLID STATE	PIN 5=+LEAD, PIN 8=-LEAD
CURRENT (22mA max)	PIN 5=+LEAD, PIN 6=-LEAD
VOLTAGE (5.5V max)	PIN 7=+LEAD, PIN 6=-LEAD



#### SENSOR INFORMATION

RTD probes:

100 ohm at 0 deg C with alphas of .385 or .392. Lead compensation is performed for 3 and 4 lead probes. Shorted and open probe detection is provided. Self heating is minimized since only 1/2 mA of probe current is used. The range supported is -200 deg C to +325 deg C.

J thermocouple:

Open probe detection is provided. Backward probe hookup is detected during CAL. Software local junction compensation is provided. The ranges supported are; narrow= -200 to 325 deg C, wide= -200 to 760 deg C.

K thermocouple:

Open probe detection is provided. Backward probe hookup is detected during CAL. Software local junction compensation is provided. The ranges supported are; narrow= -200 to 325 deg C, wide= -200 to 1250 deg C.

T thermocouple:

Open probe detection is provided. Backward probe hookup is detected during CAL. Software local junction compensation is provided. The range supported = -200 to 325 deg C

SOLID STATE:

Analog Devices AD590 type sensor support. The sensor passes approximately 1 uA per deg Kelvin. Open and shorted detection. The range supported = -60 to 160 deg C.

VOLTAGE:

Provides for any sensor that provides a voltage output that is, or can be made to be, within a 0 to 5 volt range. During the CAL menu the PC1000 will provide for linear fitting to the sensor (offset and gain cal). The source impedance of the voltage source should be 1 K ohm or less. If higher input impedance is required, contact SUN for assistance.

## CURRENT:

Provides for any sensor that provides a current output that is, or can be made to be, within a 0 to 20 mA range. During the CAL menu the PC1000 will provide for linear fitting to the sensor (offset and gain cal). The input resistance is 250 ohm. If a different input resistance is required then contact SUN for assistance.

To determine sensor input resolution for a particular sensor type, divide the total sensor range by 32767 (15 bit conversion). The PC1000 checks the temperature probes for open or shorted condition about once per minute. Therefore after a probe error has been corrected it will take up to 1 minute for the PC1000 to recognize that the probe is ok. The probes are electrically isolated for up to 240 volt. There is an equivalent capacitance of about .03 uF from the probe leads to PC1000 ground.

## ON/OFF PROCESS I/O PORT

The PROCESS bit I/O port contains the on/off output signals that are used to control the processes as well as on/off outputs and inputs that are available to the user. The PC1000 process port provides 8 output bits, 8 pullups to 5 volts, 8 input bits, and an external fail-safe input. The signal lines that are used to control the processes consist of the following. The most significant bit, bit 8, is used as a power on/off control to the process while the lower 4 bits control heat and cool for each channel. The remaining 3 bits, unless used in a special PC1000 mode, may be used for general on/off control. All 8 input bits are available to the user. The user accesses the free bits via IN and OUT commands. As shown, the process port is a 26 pin flat cable compatible connector located on the rear panel. The following is a schematic of the internal PC1000 circuitry that drives the 26 pin connector.





Note that the lower 4 bits, that are used for heat and cool control, can be inverted by cutting the appropriate jumper that is located on the internal PC1000 logic board. For example if you wish the channel 1 cool output to be off when cooling, then cut jumper 7. To access the jumpers, remove power to the PC1000, remove the cover, locate the jumpers a couple of inches below the process port connector. The jumper is then cut open to invert its output bit.

PROCESS PORT bit I/O port signals (26 pin IDC)

PIN #	DESCRIPTION
* 1	/FAIL-SAFE IN
* 2	CHAN #1 + OUT (to raise process 1) (F1)
* 4	CHAN #1 - OUT (to lower process 1) (F2)
* 6	CHAN #2 + OUT (to raise process 2) (F3)
* 8	CHAN #2 - OUT (to lower process 2) (F4)
10	for user output (F5)
12	for user output (F6)
14	for user output (F7)
* 16	POWER ON/OFF OUT (F8)
18, 20 to 26	270 ohm pull ups to +5 volts (P1-P8)
3	ground
19,17,15,13, 11,9,7,5	8 input bits (G1-G8) for user TTL or dry contact. Pullup to +5v internally connected

\* These are standard process control outputs

The standard PC1000 comes with a flat cable, to plug into the process port, and a printed circuit board that the flat cable is plugged into. The printed circuit board provides screw terminal connections to the process bit I/O lines to make connection of your system as easy as possible. The outputs are designed to directly drive solid state relays that control power to your process circuits. The PC1000 outputs are not able to drive power loads directly. Solid state relay modules can be purchased from SUN or through normal distribution. They are available with a number of voltage and current ratings. The PC1000 turns on the solid state relay (SSR) by sourcing current from the 5 volt pullup, through the SSR input light emitting diode, to ground via the open collector output. The 8 input bits are provided with 10 K ohm pull up resistors to 5 volts, allowing a simple switch closure to ground as an input, as well as being 5 volt logic compatible.

To write to one of the 3 available output bits

OUT0:m,n ;where m=4(F5), m=5(F6), or M=6(F7)  
;where n=1 for on, 0 for off

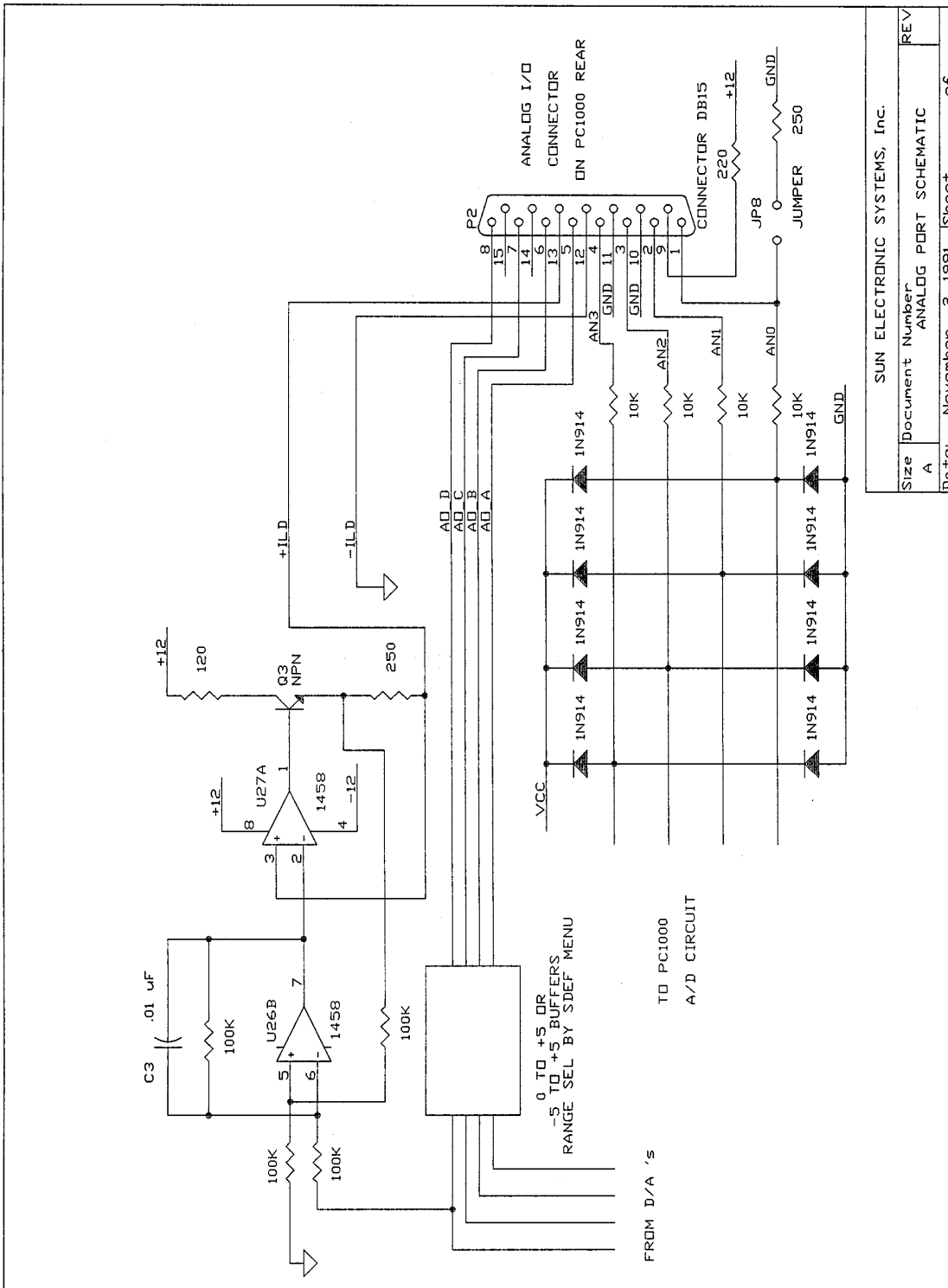
To read one of the 8 available input bits

IN0:m,In ;where m=0 to 7 for G1 to G8  
;where In is an I variable n

See the PC1000 installation section, under connecting process control outputs, for hookup examples.

## **ANALOG INPUT & OUTPUT PORT**

The analog interface consists of 4 channels of 8 bit precision analog outputs,(channel A,B,C, and D), and 4 channels of 8 bit precision analog inputs(channel A,B,C, and D). This auxiliary analog interface is useful when interfacing with additional sensors, analog instrumentation and analog control devices. Any of these analog I/O channels, that you have not configured for use via the menu procedures, may be accessed directly by using the IN and OUT commands. In the SDEF menu, analog output channels may be independently configured as uni-polar (0 to 5 v) output range or bi-polar (-5 to +5) output. In addition analog output channel D provides a 0 to 20mA current source buffer associated with it. Analog input channels accept inputs in the range of 0 to +5 volts. In addition analog input channel A may be jumper configured to accept a 20mA current input and a 220 ohm resistor tied to +12 volts is provided for supplying current to a passive source if needed. The following is the schematic of the internal PC1000 analog I/O circuitry;



Size	Document Number	SUN ELECTRONIC SYSTEMS, Inc.
A	ANALOG PORT SCHEMATIC	
Date:	November 2, 1991	Sheet of
		REV

## Analog I/O interface (DB15)

PIN #	DESCRIPTION
1	AN0 analog input channel A address 0
2	AN1 analog input channel B address 1
3	AN2 analog input channel C address 2
4	AN3 analog input channel D address 3
5	AO0 analog out channel A address 0
6	AO1 analog out channel B address 1
7	AO2 analog out channel C address 2
8	AO3 analog out channel D address 3
9	AN0+ pull up to +12 volts
10,11	GROUND
12	-IS3 20mA Source -lead
13	+IS3 20mA Source +lead
14,15	no connection

To send data to a analog output the OUT command is used as follows;

OUT3:n,m     where n=output channel address 0=A,1=B,2=C,3=D  
                  m=a value of 0 to 255

To read a analog input the IN command is used as follows;

IN3:n,Im     where n=input channel address 0=A,1=B,2=C,3=D  
                  m is a I variable number 0 to 9  
                  the I variable can then be read  
                  via the Im? command.

The SDEF menu allows you to select D/A channel A for charting channel 1 process variable, D/A channel B for charting channel 2 process variable, channel C for output of channel #1 control signal, and D/A channel D for output of channel #2 control signal. The outputs are configurable, in SDEF, for 0 to +5 volt or -5 to +5 volt

## RS232/422 REMOTE SERIAL INTERFACE

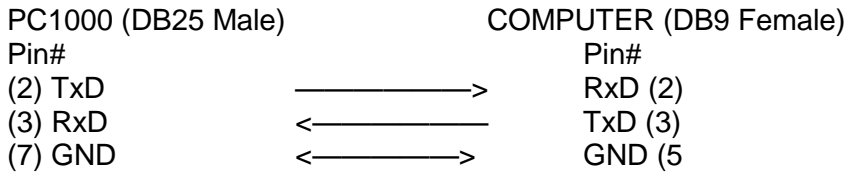
The PC1000 provides for RS232 or RS422 remote control. The primary difference between RS232 and RS422 is the type of line drivers and receivers used. RS422 has higher noise immunity and can drive much longer cables. RS232 however is the more common serial interface.

The serial port's operation depends on whether CMD ERROR INT'S are enable or disabled in the SINT menu. If command error interrupts are disabled, no indication to the host computer communicating to the serial interface is given as to whether a command was accepted or rejected. To determine last command status the ? or STATUS? command may be used. If command error interrupts are enabled, the controller responds with a CMD ERROR!! message if the command received was rejected. If the command was accepted and the type of command does not invoke a reply, i.e., such as a SET=nnn.n command, the controller will reply with an OK message. If the accepted command invokes a reply, then the reply is sent without the OK message. The CMD ERROR INT ENABLED mode is convenient since every command sent to the controller generates a reply back to the host computer thereby providing a handshake.

The RS port operating parameters are set up in the SDEF menu. The SDEF menu allows setting of the BAUD RATE, RS232 or RS422 levels, ECHO on/off, and whether to use hardware handshake. The serial port sends a zero for the parity bit and does not check incoming parity. When

DTR/CTS is not selected only the TRAN DATA, REC DATA and GROUND signal lines are significant to the controller. To communicate over the RS232 bus, a cable constructed with a male DB25 connector on the PC1000 controller end and a female DB9 connector on PC end wired as follows is required:

**RS232 Cable Wiring Diagram for PC1000**



To communicate through Windows, use the following settings:

Connect using desired COM port

Bits per second: 9600 (default setting in PC1000 SDEF menu)

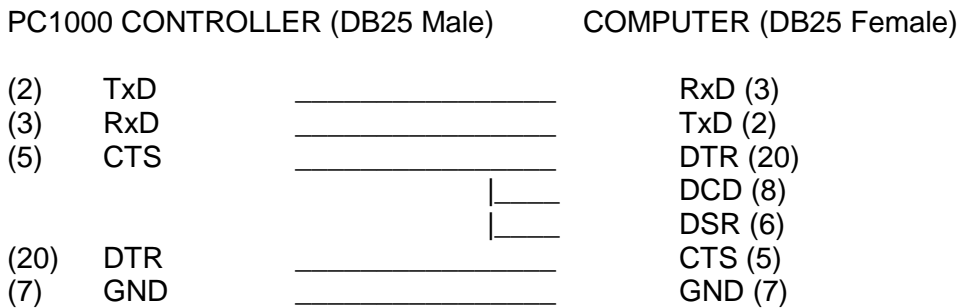
Data bits: 8

Parity: NONE

Stop bits: 1

Flow control: NONE

If your computer is using a DB25 Male connector instead of the DB9, the wiring connections are as follows. For example purposes this diagram also assumes DTR/CTS control is selected.



**\*\* Note:** CTS & DTR are Optional. If enabled in SDEF Menu but not hardwired, unit will not communicate.

The PC1000 provides RS232 or RS422 serial interface selectable by the SDEF menu. Hardware handshake signals are provided for synchronization. The signal pin assignments are as follows.

## RS232/422 serial interface (DB25 connector)

PIN#	DESCRIPTION
1,7,13,15,17	GROUND
2	TXD RS232 TRANSMIT DATA
3	RXD RS232 RECEIVE DATA
5	CTS RS232 CLEAR TO SEND
20	DTR RS232 DATA TERM READY
12	+TXD RS422 TRAN DATA +
25	-TXD RS422 TRAN DATA -
10	+RXD RS422 REC DATA +
23	-RXD RS422 REC DATA -
9	+CTS RS422 CTS+
22	-CTS RS422 CTS-
11	+DTR RS422 DTR+
24	-DTR RS422 DTR-

The following ascii interrupts are sent to the serial ports. The interrupts are followed by carriage return, linefeed. Interrupts will not be sent during the time that the PC1000 is sending a normal line of data over the interface.

### PC1000 INTERRUPT ASSIGNMENTS

---

Condition	Serial Port ASCII
SINGLE TEMP TIME OUT #1	I
SINGLE TEMP TIME OUT #2	J
COMMAND ERROR INT	CMD ERROR!!
LP TIMEOUT #1	P
LP TIMEOUT #2	Q
LP DONE	E
DEVIATION LIMIT #1	D
DEVIATION LIMIT #2	F
UPL1 EXCEEDED	O
LOL1 EXCEEDED	U
UPL2 EXCEEDED	+
LOL2 EXCEEDED	-
POWER GOING DOWN	!
POWER UP NO AUTO CONT.	Z
POWER UP AUTO CONT.	X
BKPNT	B

## IEEE-488 (GPIB) INTERFACE

The PC1000 incorporates a IEEE-488 standard interface for communication with a host computer. The PC1000 will LISTEN, TALK, SPOL, PPOL and send SRQ's. When CMD ERROR INT'S are enabled, the GPIB port sends a CMD ERROR SRQ interrupt on commands received with errors. IF command error interrupts are disabled, then no interrupt is sent to the GPIB port if command errors are detected. In all cases, the only data that is sent to the GPIB interface by the PC1000 is that which is requested by a command received by the controller. The GPIB port does not respond with "OK" after commands, that do not send back data, since handshaking is a inherent bus feature. Last command status can be determined through the use of the ? command or the STATUS? command if polling is preferred over interrupts.

The SDEF menu allows you to set the GPIB address, and the GPIB LOCKUP TIME, which is the length of time that the PC1000 will wait for the host computer to accept data before generating a GPIB LOCKUP interrupt. The SINT menu also provides the ability to set up the parallel poll bit. The GPIB interface may request or send information from/to another I/O port by issuing the appropriate IN or OUT instructions. The PC1000 GPIB signal pin assignments are as follows.

PIN #	SIGNAL	PIN #	SIGNAL
1	DIO1	13	DIO5
2	DIO2	14	DIO6
3	DIO3	15	DIO7
4	DIO4	16	DIO8
5	EOI	17	REN
6	DAV	18	GND
7	NRFD	19	GND
8	NDAC	20	GND
9	IFC	21	GND
10	SRQ	22	GND
11	ATN	23	GND
12	SHIELD	24	GND LOGIC

The PC1000 recognizes a line feed with or without EOI set, or EOI set during last character as an input line terminator. The PC1000, when talking, will send a line feed with EOI set as its line terminator. The PC1000 will pull SRQ line low if it needs to send an interrupt. If the host computer does not perform a serial poll of the PC1000 before another interrupt is generated by the PC1000, then the PC1000 will display "SRQ OVERFLOW" on the front panel. This message is to inform the user that interrupts have not been read into the host computer. The message can be ignored if you do not wish to provide an interrupt handler in your host software. Also by disabling interrupts in the SINT menu, or via the SINT= command, the interrupt may be turned off. The parallel poll bit, as set up in the SINT menu, can be used to increase the serial polling speed when there are a number of interrupting devices on the GPIB bus.

The following interrupt values are sent to the GPIB port. Many of the interrupts may be disabled via the SINT menu. IEEE-488 (GPIB) interrupt values are the value received when a serial poll is conducted.



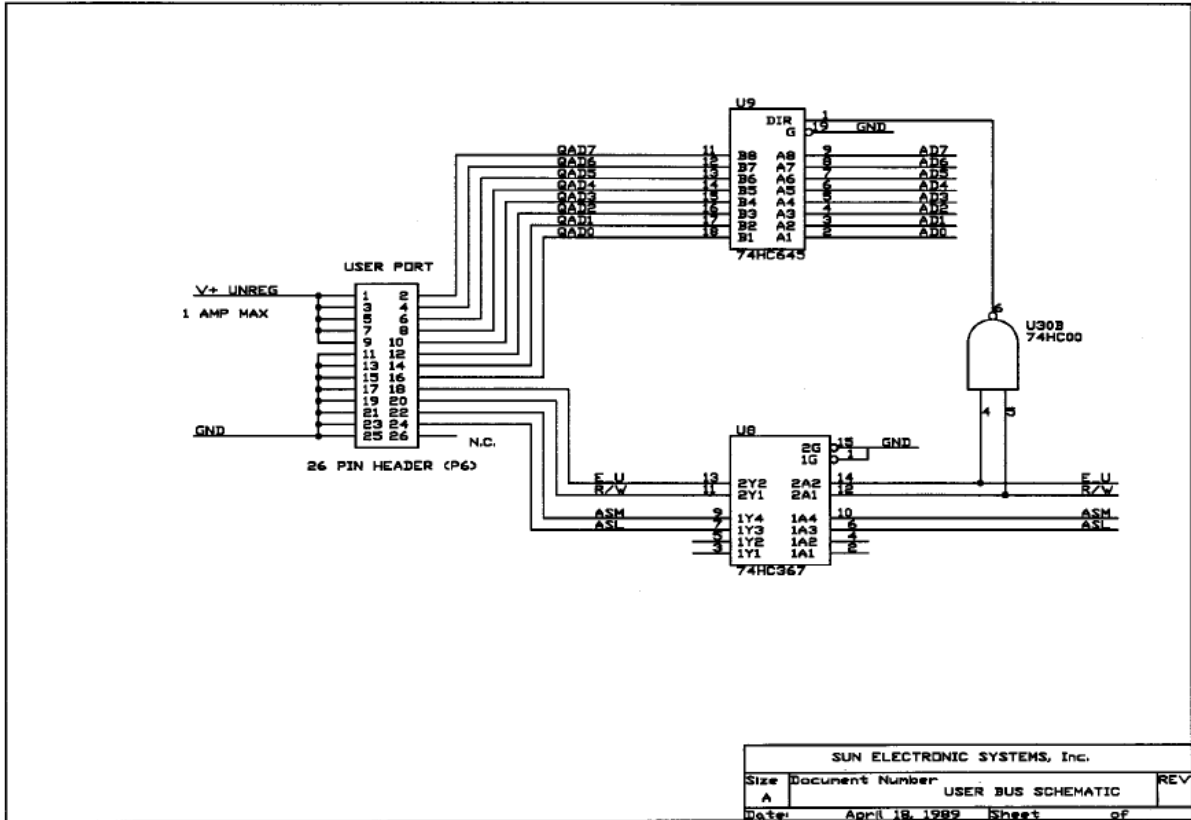
## PC1000 INTERRUPT ASSIGNMENTS

---

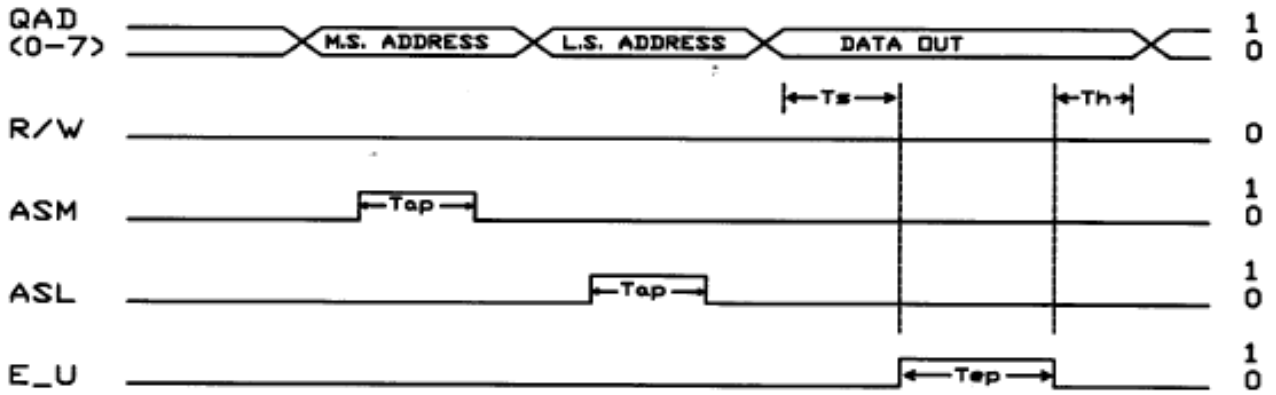
Condition	GPIB Port	
	HEX	DECIMAL
NO INTERRUPT	\$00	0
SINGLE TEMP TIME OUT #1	\$41	65
SINGLE TEMP TIME OUT #2	\$51	81
COMMAND ERROR INT	\$42	66
LP TIMEOUT #1	\$43	67
LP TIMEOUT #2	\$53	83
LP DONE	\$45	69
DEVIATION LIMIT #1	\$46	70
DEVIATION LIMIT #2	\$56	86
UPL1 EXCEEDED	\$47	71
LOL1 EXCEEDED	\$4A	74
UPL2 EXCEEDED	\$57	87
LOL2 EXCEEDED	\$5A	90
GPIB LOCKUP INT	\$4F	79
POWER GOING DOWN	(NONE)	
POWER UP NO AUTO CONT.	(NONE)	
POWER UP AUTO CONT.	(NONE)	
BKPNT	\$50	80

# USER I/O PORT

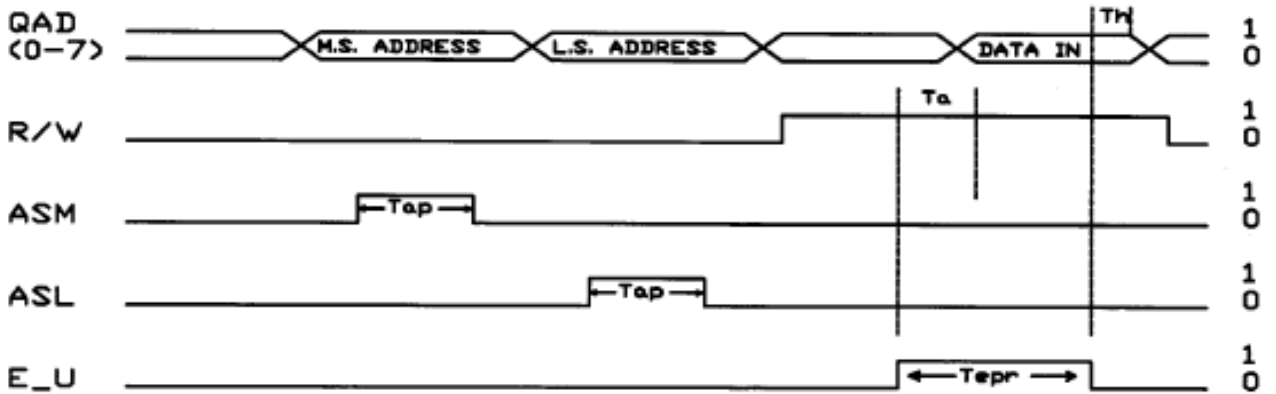
The user port is a 26 pin flat cable connector, located on the PC1000 rear panel. This byte serial multiplexed interface provides the user, after demultiplexing, with 16 bits of address and an 8 bit bidirectional data bus. Control signals for upper and lower address and data permit simple demultiplexing of the bus. The purpose of this interface is to provide the user with up to 64 k bytes of read/write address space to be used for SUN accessories or for whatever the user may require. The USER bus is accessed using IN and OUT commands. The following provides the schematic of the internal PC1000 user port circuitry, the timing of user bus signals, and a user bus example circuit.



# USER BUS TIMING (WRITE)



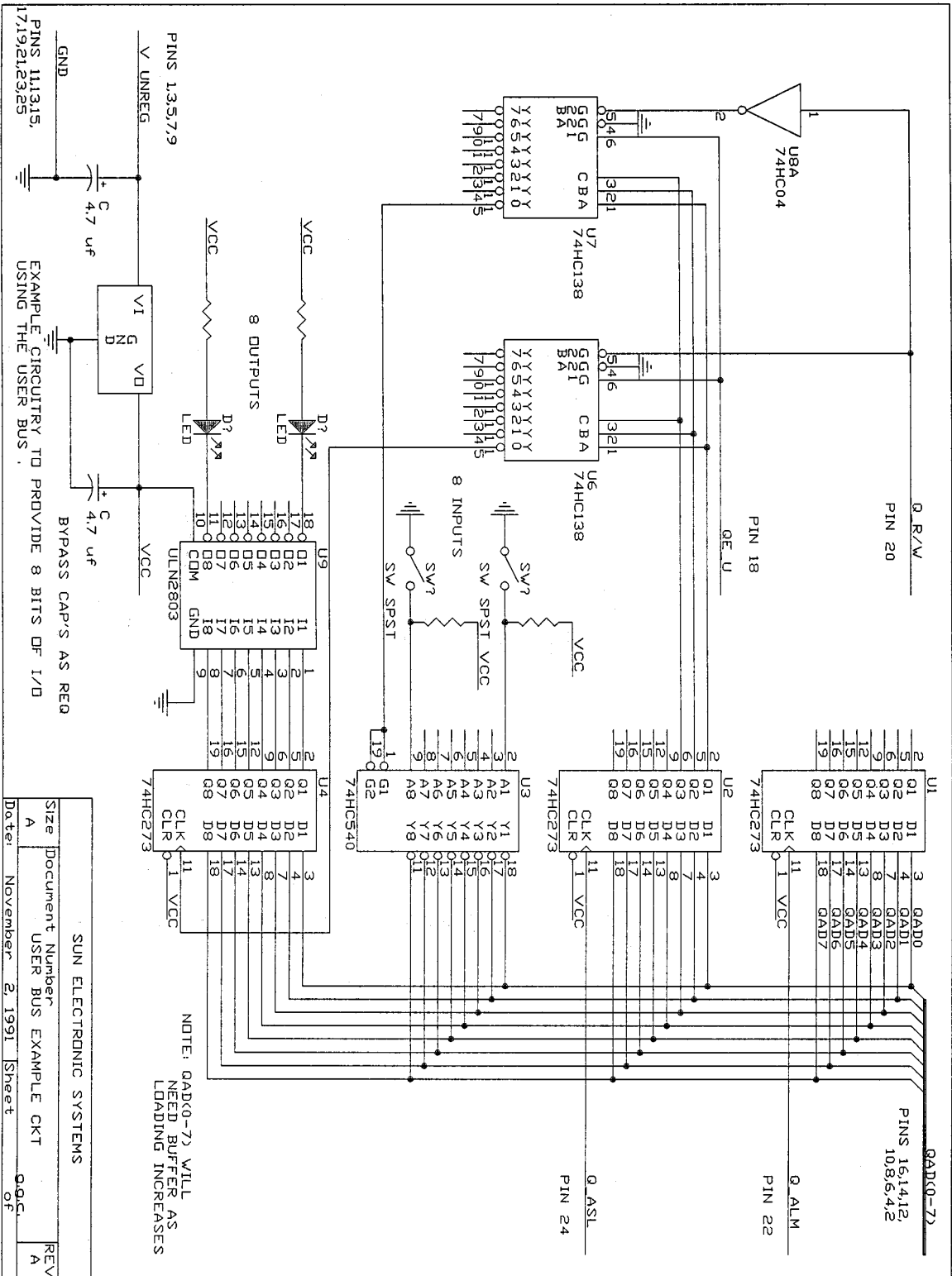
# (READ)



Ts=6  $\mu$ S MIN  
Th=3  $\mu$ S MIN  
Tap=4.8  $\mu$ S  
Tepr=6  $\mu$ S

Ta  $\leq$  3  $\mu$ S  
0  $\leq$  Th  $<$  2  $\mu$ S  
Tepr=11  $\mu$ S

SUN ELECTRONIC  
SYSTEMS Inc.



To write to the user port the OUT1:address,data command is used. To read from the port the IN1:address,lm command is used.

#### User I/O Interface (26 pin IDC)

PIN #	DESCRIPTION
2,4,6,8,10	QAD7 thru QAD0 Multiplexed
12,14,16	8 bit address/data bus
18	QE_U active high enable
20	Q_R/W read write control line
22	Q_ASM upper address strobe
24	Q_AS_L lower address strobe
26	reserved
11,13,15,17,19	GROUND
21,23,25	GROUND
1,3,5,7,9	V+ UNREG unregulated DC voltage provided to the user. approx 15 v may use up to 1 Amp max (fused)

#### HIGH SPEED SERIAL LINK INTERFACE (SPI)

This is a 56 K bit synchronous serial data link which is intended for distributed microprocessor communications. The PC1000 is configured as master device and the interface levels conform to RS422 levels. This port is reserved for future SUN SYSTEMS expansion use.

#### High speed serial link (DB9)

PIN #	DESCRIPTION
1	SCK+ serial clock +
6	SCK- serial clock -
2	MOSI+ master out slave in +
7	MOSI- master out slave in -
3	MISO+ master in slave out +
8	MISO- master in slave out -
4,5,9	GROUND



## **IV COMMAND OVERVIEW**

The PC1000 accepts commands from either the local front panel or remotely from the GPIB or RS232/422 serial interfaces. Commands from the local front panel consist of one or more keystrokes followed by depressing the enter key. Data is displayed as you type with back space and line cancel keys provided to simplify entry. Commands from the GPIB interface should be terminated with end or identify (EOI) and or line feed. Ascii commands from the RS232/422 interface may be terminated with a line feed and or carriage return. If for any reason a command is not acceptable by the PC1000, the user can be notified and the reason for the rejection can be determined. Numeric values with leading and trailing zero's and spaces are accepted and integer, fixed and floating point values are converted as required. To aid the user, all commands are as meaningful and concise as possible.

### **PROCESS CONTROLLING COMMANDS**

Process control commands allow you to set or examine process control segment parameters.

C1?	Examine present chan#1 value
C2?	Examine present chan#2 value
SET1=nnn.nn	set process control value chan #1 in current chan #1 scale of measure
SET1?	Examine present segment set point #1 in current chan #1 scale of measure
SET2=nnn.nn	set process control value chan #2 in current chan #2 scale of measure
SET2?	Examine present segment set point #2 in current chan #2 scale of measure
CSET1?	Examine instantaneous target during ramp chan #1 chan #1 scale of measure
CSET2?	Examine instantaneous target during ramp chan #2 chan #2 scale of measure
WAIT1=nn:nn:nn	wait at set point #1 (hrs:min:sec)
WAIT1=nn	wait at set point #1 (minutes)
WAIT1=F[orever]	wait1 never to time out
WAIT1?	Examine present wait time #1
WAIT2=nn:nn:nn	wait at set point #2 (hrs:min:sec)
WAIT2=nn	wait at set point #2 (minutes)
WAIT2=F[orever]	wait2 never to time out

WAIT2?	Examine present wait time #2
RATE1=nnn.nn	set ramping rate chan #1 chan #1 scale of measure per minute
RATE2=nnn.nn	set ramping rate chan #2 chan #2 scale of measure per minute
RATE1?	Examine present ramping rate chan #1
RATE2?	Examine present ramping rate chan #2
SCALE 1?	Examine present unit of measure chan #1 response is DEG C,DEG F,DEG K,RH OR U
SCALE 2?	Examine present unit of measure chan #2 response is DEG C,DEG F,DEG K,RH OR U

## PROCESS SAFETY LIMIT COMMANDS

Upper and lower limit commands guard against command setting errors and process run away conditions. Deviation limits insure that the process is where you set it.

LOL1=nnn.n	set lower limit chan #1, in scale 1 units
LOL2=nnn.n	set lower limit chan #2, in scale 2 units
LOL1?	Examine low limit chan #1
LOL2?	Examine low limit chan #2
UPL1=nnn.n	set upper limit chan #1, in scale 1 units
UPL2=nnn.n	set upper limit chan #2, in scale 2 units
UPL1?	Examine upper limit chan #1
UPL2?	Examine upper limit chan #2
DEVL1=nnn.n	set deviation limit chan #1
DEVL2=nnn.n	set deviation limit chan #2
DEVL1?	Examine deviation limit #1
DEVL2?	Examine deviation limit #2

The PC1000 will not allow you to enter a set point outside of the limits. If a process variable exceeds the upper or lower limit the PC1000 automatically turns off the appropriate process control output enable and provides an audio warning as well as interrupts to the host computer. In addition to the above mentioned limits the PC1000 provides additional safety features. An external fail safe input is provided that the user should connect to an external sensor or circuit. This input will turn off all



process control outputs when activated. Also an internal watch dog timer circuit is built in that will turn off the process outputs should the PC1000 malfunction.

## CONTROL LOOP PARAMETER COMMANDS

The ability to change the default control loops coefficients allows you to tailor the servo responses of your system.

PID1+=nn,nn,nn	set PID chan #1 process ctl +
PID1-=nn,nn,nn	set PID chan #1 process ctl -
PID2+=nn,nn,nn	set PID chan #2 process ctl +
PID2-=nn,nn,nn	set PID chan #2 process ctl -
PID1+?	examine chan #1 + PID's
PID1-?	Examine chan #1 - PID's
PID2+?	examine chan #2 + PID's
PID2-?	Examine chan #2 - PID's
PWMP=nn	set PID pulse width modulation period (2 to 30 sec)
PWMP?	Examine period

## LOCAL PROGRAM COMMANDS

The Model PC1000 provides commands to control the flow of programs which permit automatic local program operation. Up to 10 programs may be stored in the PC1000's nonvolatile memory, either from the local control panel or from the GPIB or RS232/422 interfaces. Programs may call other programs as subroutines with the GOSUB instruction. The PC1000 provides ten global integer variables designated I0 through I9. The I variables may be used in FOR In / NEXT In looping constructs for multiple pass loops, may be used in IN and OUT instructions for communication with auxiliary ports and may be used in an arithmetic expression containing additions and subtractions. A program is entered into the controller from the front panel using the EDIT n function, where n is the number of the program 0 through 9. Line editing of a stored program is also supported from the front panel. Whole programs may be uploaded to or downloaded from a remote host computer via LIST and STORE commands. A program is executed using the RUN command and prematurely terminated with the STOP command. Debug assistance is provided with the BKPNT nnn or BKPNT In instructions which permits the programmer to place breakpoints in the program to suspend program execution until the enter key is depressed on the front panel, or if run started via remote bus, until a BKPNTC command is received. BKPNT nnn displays the value nnn when encountered and BKPNT In displays the current value of variable In when encountered.

EDIT n	edit program n=0 through 9
RUN n	run program n now

RUN n TIME=n:n:n	run prog n when time of day = n:n:n
STOP	stop all program and ctl activity
DELL	delete current line
DELP n	delete program n
BKPNT nn	breakpoint constant
BKPNT In	breakpoint I variable
FOR In=ll,mm,[+/-]	FOR statement marks the beginning of a loop. ll=starting In value, mm=ending value. The In variable is either incremented [+ ,default] or decremented [-] from the starting value. FOR statements may be nested 4 deep.
NEXT In	marks the end of matching FOR loop
In=Im +/-kkk	set I variable
GOSUB n	call program n as a subroutine returns when program n end is found. May nest 4 deep.
END	program end marker
END [LP]	terminate edit session
LIST n	Up load program n to host computer
STORE N	Down load program n from host computer
BKPNT?	remote command to get bkpnt value
BKPNTC	remote command to continue from bkpnt

In addition to local program type commands, local programs may contain process controlling commands, safety limit commands, loop parameter and control commands, as well as input and output commands.

## **POWER AND CONTROL LOOP ON/OFF COMMANDS**

These commands turn the PC1000 main power on/off and enable or disable the process control loops output. On the front panel, these functions have separate function keys. The switches operate in a push-on/push-off fashion. Led indicators are provided and are on when the function is enabled. The following commands are for remote control of these functions.

ON            Turn on PC1000 main power

The ON command and the STATUS? command are the only commands that are acted on when unit is in the power off state.

Process output bit 8 is off when the PC1000 is off or in fail-safe. The output is supplied as a on/off control to your process.

OFF	Turn off PC1000 main power, also turns off process output bit.
C1ON+	Enable control loop 1 + output
C1ON-	Enable control loop 1 - output
C2ON+	Enable control loop 2 + output
C2ON-	Enable control loop 2 - output
C1OFF+	Disable control loop 1 + output
C1OFF-	Disable control loop 1 - output
C2OFF+	Disable control loop 2 + output
C2OFF-	Disable control loop 2 - output

The enable commands can be entered into local programs from the front panel, while in the edit mode, by pressing the appropriate enable/disable key. Each time the key is pressed the opposite action is provided. When a local program is run all process output enables are turned on. When the local program is finished the enables are turned off.

## AUXILIARY I/O COMMANDS

These commands provide read and or write capability to the auxiliary ports

IN0:(address),(data)	read a bit input line (address) 4 bit integer (data) 1 bit integer
IN1:(address),(data)	read the digital userport (address) 16 bit integer (data) 8 bit integer
IN2:(address),(data)	read the high speed serial link (SPI) (address) 16 bit integer (data) 8 bit integer
IN3:(address),(data)	read an analog input channel (address) 2 bit integer (data) 8 bit integer
OUT0:(address),(data)	write a bit output line (address) 4 bit integer (data) 1 bit integer
OUT1:(address),(data)	write digital user port (address) 16 bit integer (data) 8 bit integer

OUT2:(address),(data) write to high speed serial link (SPI)  
(address) 16 bit integer  
(data) 8 bit integer

OUT3:(address),(data) write to an analog output channel  
(address) 2 bit integer  
(data) 8 bit integer

The address and data fields may be a decimal or hex constant or an I variable.

## GENERAL COMMANDS

TIME=nn:nn:nn	Set time of day: hrs:min:sec [24 hr]
TIME?	Examine present time of day
TIMEE?	hours meter function
LLO	RS port local lockout front panel
RTL	RS port return front panel control
STOP	Prematurely stop LP or single
STOPE9	reboot PC1000 internal software
VER?	To determine software version
PRINT n	print a local program on optional printer
#<string>	send string to optional printer
STATUS?	poll current PC1000 status
SINT?	read selected interrupt control parameters
SINT=	set selected interrupt control parameters
SDEF?	read selected default parameters
SDEF=	set selected default parameters
?	Get status of last command always gives a 2 line response

If previous command caused an error then the offending command line and a pointer to the first offending symbol are returned. Example: you send RATT=27 command, a command error is generated if enabled in SINT menu. To determine reason for error you send ? command. The 2 line response would be

```
RATT1=27 <<
```

```
^
```

The ^ character points to the point where syntax was found to be incorrect (expected E got T). If the reason for the command error was not syntax related then an explanation for the error is returned in line 2. For example: if the PC1000 is set for single loop control, via SDEF menu, and you issue a SET2=63.4 command the 2 line response to the ? command would be

```
SET2=63.4
```

```
NOT IN DUAL MODE
```

### **Model TC01 Compatibility Commands**

The Model PC1000 provides limited upward compatibility with our Model TC01 temperature controller and our Model EC01 and EC02 environmental chambers. Consult your TC01 manual for a complete definition of these commands.

nnn.nC	Set Chamber Temperature (deg C)
	Examine Chamber Temperature (deg C)
nnn.nM	Set Time at Temperature, Wait time (minutes)
M	Examine Time at Temperature (minutes)
T	Examine Chamber Temperature in deg C always
nnnUTL	Set Upper Temperature Limit (deg C)
UTL	Examine Upper Temperature Limit (deg C)
!sss...sss	Transfer String sss...sss received from GPIB to RS-232/422 interface
S	Transfer next string received from RS-232/422 interface to GPIB



## V DETAILED COMMAND AND MENU INFORMATION

### **BKPNT n** **BKPNT Im**

Function: Local program breakpoint or pause command

Syntax:        BKPNT n        display number n at breakpoint  
                                  n is an integer number -32767 to +32767  
                  BKPNT Im     display I variable m value at breakpoint  
                                  m = 0 to 9

#### Description:

The breakpoint command allows for local program operation to be suspended until the ENTER key is depressed on the front panel or until the remote controller issues a BKPNTC command. When the BKPNT command is encountered in the local program, the buzzer sounds, its value is displayed on the front panel, and a remote bus interrupt is generated if enabled. The buzzer will periodically beep to remind you that the local program is suspended. The BKPNT value can be determined remotely using the BKPNT? command.

This command permits program debugging by providing convenient pauses in local program execution and is also useful when a host controller wishes to perform I/O at a point in the local program. The BKPNT n command displays the n number on the front panel display when encountered which is useful for identifying different breakpoints within a program. The BKPNT Im form of the breakpoint command displays the value of the I variable m which is useful for breakpoints that are placed within FOR/NEXT loops. The breakpoint command may also be entered manually from the front panel to pause a running local program.

See Also: BKPNT?, BKPNTC

Example use in a local program:

Local Program;

```
BKPNT 10
FOR I2=0,5
BKPNT I2
NEXT I2
END
```

When the local program is run BKPNT 10 be executed once then the BKPNT I2 will be executed 5 times as the program loops. The following breakpoint values will be generated:

10, 0, 1, 2, 3, 4

## **BKPNT?**

Function: To read the BKPNT value.

Description:

The BKPNT? command is a remote only command that causes the current breakpoint value to be sent to the active remote control port. If the local program is not waiting at a breakpoint a value of 0 is sent. For local operation the breakpoint value is automatically displayed on the LCD.

Restrictions: remote only

## **BKPNTC**

Function: To continue local program execution when it is waiting at a breakpoint.

Syntax:

BKPNTC

Description:

The BKPNTC command is a remote only command that causes the local program to continue execution from a breakpoint. To continue a local program from the front panel, the blank line enter key is used.

Restrictions: remote only

## **C**

Function: Set or examine the current set temperature in deg C

Syntax:            nnn.nC                    set current set1 temperature  
                      C                        examine current set1 temperature

Description:

The C command is a TC01 controller compatible command which either sets the current set1 temperature or examines the controllers current set1 temperature. The set or response is in degrees centigrade always. The C command returns with a value of -1999 if the current set temperature is equal to none. The SET1 command is more powerful and should be used rather than the C command when possible.

Restrictions: Remote interface command only.

Example:            150.0C            Set current set1 temperature to 150.0°C  
                      C                        Send command to examine current set1 temp.  
                      150.0                        Controller response, SET1 temp. = 150°C  
                      SET1?                        Examine current set1 temp with SET1 command  
                      150.0                        Controller response



## C1? & C2?

Function: Examine current Ch1 and Ch2 probe temperatures

Syntax: C1?, C2?

### Description:

The C1? and C2? commands provide the means for a remote interface to read the present value of the probes. The Ch1 value is normally displayed on the first line and Ch2 value is displayed on the second line of the front panel LCD display. The units for the probe are set in the CAL MENU for deg F, C, or K.

Restrictions: Remote interface command only.

See Also: T

Example:	C1?	read probe Ch1
	25.0	controller response, 25.0 units
	C2?	read probe Ch2
	25.0	controller response, 25.0 units



DESIRED SCALE? 1=C,2=F,3=K	Enter number to select scale for the current channel.
If voltage or current was selected;	
LOW SCALE INPUT [Yes/No]?	Apply low voltage or current to input and enter Yes when ready. Enter NO to use default 0 volt cal.
LOW SCALE VALUE ?	Enter the number that represents the process.
HIGH SCALE INPUT [Yes/No]?	Apply high voltage or current to input and enter Yes when ready. Enter NO to use default 5 volt cal.
HIGH SCALE VALUE ?	Enter the number that represents the process.
CAL CHAN 2 [Yes/No]?	Repeat similar process as above channel 1.

If a no answer was given to “CAL PROBES” prompt, then you are asked if you wish to change the default units of measure for each channel.

When performing calibration of thermocouple probes, use of a thermocouple simulator is possible if the following is considered. The PC1000 performs local junction compensation in software by reading the temperature of the sensor input connectors then calculating the effective local junction voltage at that temperature. This means that unlike the thermocouple millivolt tables that are referenced to 0 deg C, the millivolt input that the PC1000 will receive will be referenced to whatever temperature the sensor connector is at. For example, if the connector is at 25 deg C then the PC1000 will expect 0 mV input for a thermocouple temperature of 25 deg C. Also the PC1000 input has circuitry at the probe input for open thermocouple detection. If the output impedance of the thermocouple simulator is high the millivolt output of the simulator will be affected. When calibrating RTD probes resistors may be used if desired.

If a NO answer is given to force the use of the approximate default values, remember that the calibration will only be approximate. To insure accurate sensor readings fully calibrate the sensor or follow up with cal to a standard. Each sensor input is electrically isolated for up to 240 volts. The PC1000 performs a check for open/short temperature probe about once each minute. Therefore if there should be a probe error then you correct the problem it will take up to 1 minute for the probe error indication to stop.

## CAL TO A STANDARD

Cal to a standard allows calibration to your house standard instrument. Three commands are supplied to perform the calibration. "CALA" performs an offset cal to the first temperature point while "CALB" performs a gain cal between the first point and the second point supplied with the CALB command. "CALC" clears both the offset and gain cal adjustments.

To perform the cal to a standard:

1. Clear existing standard cal if any  
CALC,access code
2. Place your standard probe as close as possible to the probe or probes to be calibrated.
3. Set the process to go to the first calibration point.
4. Allow time for process stabilization.
5. To cal the channel 1 probe offset;  
CALA1,access code,standard reading
6. To cal the channel 2 probe offset;  
CALA2,access code,standard reading

where; access code is the CAL menu access code number (default=9) and standard reading is the temperature reading that your house standard instrument is reading.

7. Set the process to go to the second calibration point. The larger the difference between the two points the more accurate the gain cal will be.
8. Allow time for process stabilization.
9. To cal the channel 1 probe gain;  
CALB1,access code,standard reading
10. To cal the channel 2 probe gain;  
CALB2,access code,standard reading

## CHANNEL 1 & 2 OUTPUT ENABLES

### **C1ON+ C1OFF+ C1ON- C1OFF- C2ON+ C2OFF+ C2ON- C2OFF-**

Function: Disable or enable process outputs

Syntax: C1ON+ enable channel #1 heat or raise output  
C1OFF+ disable channel #1 heat or raise output  
C1ON- enable channel #1 cool or lower output  
C1OFF- disable channel #1 cool or lower output  
C2ON+ enable channel #2 heat or raise output  
C2OFF+ disable channel #2 heat or raise output  
C2ON- enable channel #2 cool or lower output  
C2OFF- disable channel #2 cool or lower output

#### Description:

These remote commands permit a remote interface to enable or disable the process control outputs. Front panel switches are provided for this function. When editing local programs from the front panel keyboard, depressing the enable switches on the front panel enters its function either ON or OFF. If the opposite action is desired, cancel the entry and press the switch again to get the correct action. The current state of the enables can be determined remotely by the use of the STATUS? command. If probe or limit errors should occur, then appropriate process outputs will automatically be disabled.

Restrictions: None

See Also: CON, HOFF, HON

Example: C1OFF+ disable channel 1 process heating

## **CSET1? & CSET2?**

Function: Examine the present control value during ramping.

Syntax: CSETn? where n=1 or 2

Description: The CSET command allows for remote determination of the current set point, that the controller has calculated, during ramping. The CSET values are also automatically displayed on the front panel

Restrictions: Remote only command; cannot store in a LP.

See Also: SET1?, SET2?, C1?, C2?

Example: assumes channel 1 temperature control in degrees C

RATE1=10	define temperature segment, RATE=10°C/minute
WAIT1=30	wait=30 min.
C1?	examine present channel 1 temperature
25.0	controller response, 25.0°C
SET1=35.0	set new final temperature, 35.0°C
	wait 30 seconds
CSET?	examine present ramp target temperature
30.0	controller response, 30.0°C

## **DELL**

Function: Delete current line during local program edit

Syntax: DELL

Description: The delete line (DELL) command allows for deleting program lines during local program editing on the front panel keyboard. The line currently displayed will be deleted when the DELL command is entered. To insert a line before the currently displayed program line just enter the new line via enter key.

Restrictions: Local front panel command only

See Also: DELP

Example:

EDIT#2	edit local program #2
GOSUB#0	controller response, first line of LP
[DOWN-ARROW]	use red up/down arrows for scrolling LP
BKPNT1	controller response
DELL	delete breakpoint command
GOSUB#5	controller response, next line of program
END	to exit edit mode

## DELP

Function: Delete or clear local program

Syntax: DELPm or DELP#m

Description: The delete program command allows for deleting, which in this case is the same as clearing, a single local program defined by the variable "m", where m is in the range of 0 to 9. Programs may be deleted either from the local front panel or from a remote interface. Before using the STORE command from a remote interface, the local program selected, 0 thru 9, must be deleted if any commands are present in that particular local program.

Restrictions: cannot store in LP

See Also: DELL, STORE

Example: from a remote interface  
DELP#0 clear local program #0  
STORE#0 prepare controller for accepting commands  
nnn free bytes available number is received  
\*\*\* send local program commands  
END terminate local program download

## DEVL1 & DEVL2

Function: Set or examine the deviation limit

Syntax:       DEVL1=nnn.n       Set deviation limit channel 1  
              DEVL?            Examine deviation limit channel 1  
              DEVL2=nnn.n       Set deviation limit channel 2  
              DEVL2?            Examine deviation limit channel 2

Description: The DEVL command sets the deviation limits for its channel. If the deviation limit interrupt is enabled in the SINT menu and the absolute value of the difference between the SET and measured process varies by more than the deviation limit, an interrupt will be generated to the active remote interface. The deviation interrupt will continue to be generated, about every 2 seconds, as long as the limit is exceeded. The STATUS command can also be used to determine if a deviation limit has been exceeded if you wish to poll rather than provide for interrupts.

Restrictions: Deviation Limit range is between 0.1 and 300.

See Also: UTL, LTL, SINT Menu

Example:       assume controlling temperature in deg C  
              DEVL1=2.5        set deviation limit 1 to 2.5°C  
              DEVL1?        examine deviation limit 1  
              2.5            controller response, 2.5°C  
              RATE1=30        define temp. segment  
              WAIT1=03:00:00   wait for 3 hours  
              SET1=-55.0       set final temp. to -55.0°C  
                              if after 1 hour, the process runs out of  
                              coolant supply and process temperature  
                              starts to rise, and if deviation  
                              interrupts are enabled in the SINT menu,  
                              a deviation interrupt will be sent to  
                              the active remote interface.



## EDIT

Function: Edit local program from the local keyboard.

Syntax:       EDIT m or EDIT #m

Description:   The EDIT command allows for program loading and editing from the local keyboard. A local program, specified by m in the range of 0 to 9, may be edited by using the up and down arrows, delete line (DELL) command or by entering new commands which are placed in front of the command presently displayed on the front panel LCD display. Local program editing is terminated by entering the END command.

Restrictions:  Local keyboard command only.

See Also:     STORE, LIST, END

Example:      This example illustrates entering and editing a local program from the front panel keyboard. The local program consists of a call to LP#2, a breakpoint which displays the number 1, and a call to LP#5.

DELP 0	to clear existing LP if any
YOU SURE [Y/N]	
YES	You are sure you wish to delete
EDIT #0	edit local program #0
END	controller response on LCD display
GOSUB #2	enter prog. call to local prog #2
END	controller response
GOSUB#5	enter prog. call to local prog #5
END	controller response
[UP ARROW]	scroll up
GOSUB#5	controller response
BKPNT 1	insert breakpoint between calls to LP#2 and LP#5
GOSUB#5	controller response
END	enter end local program editing command

## END

Function: Terminate local program editing or remote local program downloading.

Syntax: END

Description: The END command terminates program editing when editing local programs from the local keyboard and terminates remote program downloading when using the STORE command from a remote host computer. On the front panel the end command is the red shifted END key. When the key is pressed END LP is displayed to indicate that this command will end your LP editing.

Restrictions: None

See Also: EDIT, STORE, STOP

## FOR/NEXT

Function: Loop definition command for local programming

Syntax: FOR Im = <Beginning Value>,<Ending Value> or  
FOR Im = <Beginning Value>,<Ending Value>, + or  
FOR Im = <Beginning Value>,<Ending Value>, -  
NEXT Im

**Note:** The I variable number m must match in the FOR/NEXT

Description: The FOR/NEXT loop command provides looping capability for local program operation. Commands contained within the boundaries between the FOR statement and the NEXT statement are repeated as defined by the FOR statement. The beginning or ending value may be an integer value or an I variable value. The FOR/NEXT loop either increments the beginning value until the beginning value is equal to or greater than the ending value or decrements the beginning value, when the minus option is used, until the beginning value is equal to or less than the ending value. The number of loop executions is equal to the absolute value of the ending value minus the beginning value. The loop counter is updated and checked when the matching NEXT instruction is executed therefore the loop is always executed one time. FOR/NEXT loops may be nested four deep when considering the first loop as nest#1.

When a local program is running or has been terminated due to an error, by using the Im? command, it is possible to determine where the program is or was in its execution.

Restrictions: Local program command

See Also: In?, In=, BKPNT In

Example: FOR I0=0,7            I0 will be the loop counter variable  
                                 it will start equal to 0. Default  
                                 increment mode is assumed. The value  
                                 7 is saved as the termination value.

“other commands”

NEXT I0	I0 will be incremented and if it is less than 7 the program will go back up to the FOR instruction
---------	--

Example remote store:

DELP#2	clear program #2
STORE#2	load local program 2 via remote bus
FOR I5=1,5	FOR I variable #5, step 1,2,3,4
FOR I2=5,I5,-	FOR I variable #2,decrement from 5 to the current value of I variable #5. The operation will be: step 5,4,3,2 (Pass 1) step 5,4,3 (Pass 2) step 5,4 (Pass 3) step 5 (Pass 4)
BKPNT I2	set breakpoint I variable #2
NEXT I2	End of FOR I2 loop
NEXT I5	End of FOR I5 loop
END	end program store
RUN#2	run program #2

when the program is run, the breakpoint value will display the step sequence for the FOR/NEXT I2 variable loop.

## GOSUB

Function: Execute local program “m” as a subroutine

Syntax: GOSUB m or GOSUB #m

Description: The GOSUB command allows local programs to be called by other local programs as subroutines. At the end of a local program called by a GOSUB command, execution returns to the command following the GOSUB command in the calling program. Nesting is supported for four levels considering the first calling program as nest 1. I variables are global to all local programs.

Restrictions: Local program command

See Also: FOR/NEXT

Example: Listing of local program 2

FOR I3=0,10	set FOR/NEXT for 10 cycles
GOSUB#0	execute LP#0
GOSUB#1	execute LP#1
NEXTI3	end FOR/NEXT loop for I variable #3
END	end of LP#2

## I Variables

Function: Set or examine I variables.

Syntax:

I <sub>m</sub> = <value>	set I <sub>m</sub> equal to value {-32767 to 32768}
I <sub>m</sub> = I <sub>m</sub> + <value>	set I <sub>m</sub> equal to sum of values
I <sub>m</sub> = I <sub>m</sub> - <value>	set I <sub>m</sub> = to difference of values
I <sub>m</sub> ?	examine I variable m

Description: I variables are used in local programs for loop control, for input values read by IN commands, variable data for OUT commands, and general purpose Integer arithmetic. A value may be any integer in the range of + 32768/-32767 or another I variable. The controller supports ten I variables, where “m” ranges from 0 to 9. I variables are global between all local programs and as such, care should be taken to insure that the same I variable is not used in two different applications within any active group of local programs.

Restrictions: None

See Also: FOR/NEXT, IN, OUT, BKPNT

Example:	I0=52	set I variable #0 to 52
	I2=I0	set I variable #2 equal to I variable #0
	I5=I0-9	set I var.#5 = I var.#0- 9 = 43
	I6=I0+I5	set I var.#6 = I var.#0 + I var.#5 = 95
	I6?	examine I variable #6
	95	controller response
	IN0:1,I3	read an input and store in I3

## IN

Function: Input data from the PROCESS port, USER bus, SPI port, or A/D port

Syntax: IN<device number>:<address>,I<sub>m</sub>

Description: The IN command allows data read capability from either the PROCESS port, USER bus, SPI port or analog to digital convertors. Data values read from a particular device are stored in I variables. For proper operation of each device interface consult the appropriate section describing the device in full. The IN command may be stored in local programs.

Restrictions: None

See Also: OUT, I variables

Example:	IN0:5,I2	Read AUX bit 5, store in I var.#2
	IN1:7,I2	Read USER bus, address 7, store in I var.#2
	IN1:\$F,I2	Read USER bus, addr. HEX F, store in I2
	IN2:I3,I2	Read SPI port, addr in I3, store in I2
	IN3:2,I2	Read A/D channel 2 into I var.#2

Example: using IN to wait for external input high state when running a local program.

```
FOR I1=0,2
  IN0:0,I1          read process bit 0 into I1 . When bit 0 is high, I1 will equal 1
                   allowing the loop to end
NEXT I1
```

## LIST

Function: List a local program to a remote interface

Syntax: LIST m or LIST #m

Description: The LIST command permits a remote interface to upload a local program from the controller. This command allows a remote operator to verify the results of a STORE instruction, which downloads programs to the controller. In addition, SUN's RS232 demo program and some communication programs allow for capturing data to disk files. In this manner, local programs can be maintained on a personal computer and downloaded or uploaded if required.

Restrictions: Remote interface command only

See Also: STORE n, PRINT n

Example: DELP#0 clear local program #0  
STORE#0 ready controller for local program download  
1237 controller sends amount of free memory  
  
\*\*\* send local program  
  
END« end download  
LIST#0request upload of LP#0  
  
\*\*\* local program sent to remote bus  
  
END end of listing

## LLO

Function: RS-232/422 Command to lockout local keyboard

Syntax: LLO

Description: The Local Lock Out (LLO) command may be issued from a host processor to lockout the local keyboard. The LLO command only applies to host computers communicating over the RS-232 or RS-422 serial interfaces. Local keyboard function may be unlocked thru the use of the RTL command. Local keyboard lockout operation for host processors communicating over the IEEE-488 interface is a bus interface command. The syntax varies according to the particular host computer software.

Restrictions: RS-232/422 remote interface command only.

See Also: RTL

Example: LLO lockout local keyboard  
\*\*\*\*\* run sensitive testing  
RTL unlock local keyboard

## LOL1 & LOL2

Function: Set or examine the lower limit

Syntax: LOL1=nnn.n Set low limit for channel 1  
LOL1? Examine low limit channel 1  
LOL2=nnn.n Set low limit for channel 2  
LOL2? Examine low limit channel 2

Description: The LOL command allows for setting the lowest process set value that the PC1000 will accept as a valid setting. If the process should ever goes below the LOL value, the cool or decrease enable for that channel will be disabled and an interrupt to the active remote will be generated. The assumed units for the LTL command are those as selected for its channel, as set in the CAL menu.

Restrictions: The range of values for the LOL commands are determined by the type of sensor selected for its channel.

See Also: UPL1, UPL2

Example: assume RTD sensor channel 1 °C as scale  
LOL1=-100.0 set LOL to -100°C  
LOL1? examine LOL channel 1  
-100.0 controller response, -100°C

## M

Function: Set or examine channel 1 wait at setpoint time.

Syntax: nnn.nM Set channel 1 segment wait time  
M Examine the current channel 1 wait time

Description: The M command is a TC01 controller compatible command which either sets the current channel 1 segment wait time, sometimes referred as soak period, or allows a remote interface to examine the current setting. When setting wait times, nnn.n represents the wait time in minutes. If wait time is equal to forever, M returns 19999. For new applications use the more general WAIT1 command.

Restrictions: Remote command only.

See Also: WAIT1

Example: 12.1M Set wait time for 12.1 minutes  
M Examine wait time  
12.1 Response from controller  
WAIT? Examine wait time with WAIT command  
00:12:06 Response from controller

## **OFF**

Function: Turn off PC1000

Syntax: OFF

Description: The OFF command turns off the PC1000. As long as the PC1000 is connected to an appropriate line voltage, the controller is still running but appears off. The OFF command turns off process outputs, the front panel display, most front panel indicators, and front panel switch functions, except for the POWER switch. In addition, commands received over the remote interfaces are ignored except for STATUS and the ON command. The OFF command is equivalent to depressing the front panel POWER switch when the PC1000 is ON. When the PC1000 is off, all process control outputs are off.

Restrictions: Remote interface command only

See Also: ON

Example: OFF Turn OFF controller

## **ON**

Function: Turn on PC1000

Syntax: ON

Description: The ON command turns on the PC1000. When the controller is plugged in to an appropriate line voltage, the controller is waiting for a power on command from either the front panel by depressing the power switch or from a remote interface by using the ON command. When the controller is turned on, process control output bit 8 turns on. The RUN AT TIME OF DAY will also automatically turn the PC1000 on.

Restrictions: Remote interface command only

See Also: OFF

Example: ON Turn on chamber power



## OUT

Function: Output data to the PROCESS port, USER bus, SPI port, or A/D port

Syntax: OUT<device number>:<address>,<data>

<device number> integer from 0 to 3 where;  
0=process port, 1=user port, 2=SPI, 3=analog  
<address> integer or In variable  
<data> integer or In variable

Description: The OUT command permits data to be transferred to an output device such as the PROCESS port, USER bus, SPI port, or digital to analog convertors. For a complete description on operating with an I/O port, consult the particular port section in this manual. The OUT command can be stored in local programs allowing for unique control.

Restrictions: None

See Also: IN

Example:

OUT0:5,1	turn on process output bit 6
OUT1:7,0	output data=0 to address=7 to the USER bus
OUT1:\$20,\$0F	output HEX F to addr. HEX 20, USER bus
OUT2:I2,I3	output I var#3 to addr. Ivar#2, SPI port
OUT3:2,\$80	output HEX 80 to channel C, D/A port

## PID Coefficient Modification

Function: Set or examine PID coefficients

Syntax:	PID1+=nnn.n,nnn.n,nnn.n	set PID coefficients channel 1 heating or increasing
	PID1+?	examine PID1+ coefficients
	PID1-=nnn.n,nnn.n,nnn.n	set PID coefficients channel 1 cooling or decreasing
	PID1-?	examine PID1- coefficients
	PID2+=nnn.n,nnn.n,nnn.n	set PID coefficients channel 2 heating or increasing
	PID2+?	examine PID2+ coefficients
	PID2-=nnn.n,nnn.n,nnn.n	set PID coefficients channel 2 cooling or decreasing
	PID2-?	examine PID2- coefficients

### Description:

The PIDC command permits modification and examination of the proportional, integral and derivative (PID) coefficients used by the process control algorithm. There are four sets of coefficients, or two sets for each channel. One is used when increasing the process, the other when decreasing. PID coefficients may be changed to tune the PC1000 when required. See PID note in rear of this manual for a procedure for tuning PID coefficients for your particular application. PID coefficients are stored in battery backed memory. These commands may be stored in local programs to allow PID modification on the fly.

Restrictions: None

See Also:PIDA, PWMP

Example:	PID1+=0.5,0.25,1e-3	set PID coefficients
	PID1+?	examine PID coefficients
	0.250	controller response, P
	0.001	controller response, I
	0.100	controller response, D

## PIDA

Function: To place PC1000 in advanced PID mode

Syntax: PIDA=n[,mmm.m]                      Set advanced mode n with optional damping mmm.m

n=0;	Normal PID control
n=1;	Average mode
n=2;	Slowly force channel 2 probe to SET1
n=3;	Use probe 2 as channel 1 sensor
n=4;	Control to average and force chan 2 to SET1

mmm.m;                      When above n is 2 or 4 a damping coefficient may be entered to control loop speed. damping range = 0 to 1000 default=400

PIDA?                      determine current mode

Description: The PIDA command is used to select one of SUN's advanced pid control modes. This command is valid only when the PC1000 is in its single channel control mode. When mode 2 or 4 is selected a damping coefficient may be set. If you don't specify a damping the default value is 400. The larger the damping number the slower the controller will move the channel 2 reading to channel 1 set point. The PIDA modes also change when the PC1000 starts to count down WAIT1 times, as follows;

PIDA=0	when $ABS(SET1 - CH1) < \text{trigger } 1$
PIDA=1	when $ABS(SET1 - (CH1 + CH2)/2) < \text{trigger } 1$
PIDA=2	when $ABS(SET1 - CH2) < \text{trigger } 1$
PIDA=3	when $ABS(SET1 - CH2) < \text{trigger } 1$
PIDA=4	when $ABS(SET1 - CH2) < \text{trigger } 1$

where trigger 1 is the value set up in SDEF menu

Restrictions: PC1000 must be in single channel control mode

See Also: The PID appendix for further information

Example: PIDA=4,500    Set mode 4 with 500 damping

## PRINTn

Function: Print a local program on printer

Syntax:                      PRINT n                      where n=0 to 9

Description: This command allows printing of a local program on the optional PC1000 line printer. If the optional printer is not installed then this command is ignored. The front panel red shifted OUT key is used for this command.

Example:                      PRINT 2                      print local program 2 on line printer

## PWMP

Function: Set or examine the pulse width modulator period

Syntax: PWMP=nnn set PWM period  
PWMP? examine PWM period

Description: The PWMP command permits modification of the period allocated to the pulse width modulation algorithm. In the control of the process, the controller uses a constant period of time of which the controller calculates the portion of time for which heat/increase or cool/decrease should be turned on. This is based on the feedback from the process sensor and the PID coefficients. Shorter periods of time for pulse width modulation translates into faster tracking of changes in the process. When changes in the process are expected not to change for an extended time, during long soak periods for example, it may be desirable not to adjust quite so frequently. In this case, increasing the PWMP period, reducing the modulation rate, may conserve power or reduce wear on coolant supply valves, etc. Note that longer PWMP times may reduce control loop stability.

Restrictions: Period range from 2 seconds to 60 seconds

See Also:PID appendix

Example:	PWMP=2	set PWM period to 2 seconds
	C1?	examine current temperature
	25.0	controller response
	RATE1=10	define new temperature segment
	WAIT1=5	set wait for 5 min., stabilize mass
	SET1=35.0	set new temp to 35.0
		after 1 min for ramp and
		after 5 min for wait timeout
	PWMP=15	set PWM period for 15 sec.
	WAIT1=10:00:00	set long wait period, 10 hrs.
	SET1=35.0	again, set temperature

## RATE1 & RATE2

Function: Set the ramping rate for channel 1 and 2

Syntax:

RATE1=nnn.n	Set the ramping rate of channel 1
RATE1?	Examine the current ramping rate chan 1
RATE2=nnn.n	Set the ramping rate of channel 2
RATE2?	Examine the current ramping rate chan 2

Where nnn.n may range from .001 to 1000

Description: The RATE command allows for setting or examining the rate at which the controller will linearly control the process change from one set point to another set point. When setting or examining the rate, nnn.n represents units change per minute. Units may be °F, °C, °K, % RH, or USER units of measure, as set up for the channel sensor, in the CAL menu. Rate values as set are maintained until they are changed by another rate command. If you wish to move the process from one set point to another as fast as the process dynamics will allow, then enter a large rate value. During the ramp to the new set point, the PC1000 will display the linearly calculated CSET or current set point target value. The PC1000 calculates a new point on the linear ramp "CSET" once every 2 seconds.

Restrictions: Process dynamics limit the maximum ramping rate.

See Also: SET, WAIT

Example: If SCALE for channel 1 probe is in °C

RATE1=10.0	set ramping rate for 10.0°C/minute
RATE1?	examine present ramping rate
10.0	controller response, 10.0°C/min.
C1?	read present process temperature
25.0	controller response, for example 25.0°C
SET=35.0	set new setpoint, for example 35°C

\*\*\*\*

after waiting 30 seconds

C1?	read present process temperature
30.0	response, up 5°C in one half min.

## RTL

Function: RS-232/422 Command to unlock local keyboard

Syntax: RTL

Description: The Return to Local (RTL) command may be issued from a host processor to unlock the local keyboard. The RTL command only applies to host computers communicating over the RS-232 or RS-422 serial interfaces. Local keyboard function may be locked thru the use of the LLO command. Local keyboard unlock operation for host processors communicating over the IEEE-488 interface is a standard GPIB function and varies according to the particular host computer software syntax requirements.

Restrictions: RS-232/422 remote interface command only. GPIB interface host computers have unique methods for returning local operation.

See Also: LLO

Example:	LLO	lockout local keyboard
	****	run sensitive testing
	RTL	unlock local keyboard

## RUN

Function: Starts the execution of a local program

Syntax: RUN m or RUN #m

Where n = 0 to 9

Description: The RUN command initiates the execution of local program defined by m, where m selects a local program in the range of 0 to 9. Local programs are entered into the controller using the EDIT command from the local keyboard or downloaded from a remote host computer using the STORE command. Local program execution may be suspended via a BKPNT command or prematurely terminated by issuing a STOP command. Local program execution will also be terminated if an error condition, such as a limit error or probe error, is encountered during execution.

Restrictions: None

See Also: RUN-TIME, EDIT, STOP, STORE, END

Example: this example downloads a simple program from remote processor and then executes the program. Scale in °C.

DELP#0	delete current program 0
STORE#0	prepare controller for entering local prog.
1121	free memory response
FOR I2=0,10	set FOR loop for 10 loops
RATE1=10	define temp segment, RATE1=10°C/m
WAIT1=15	wait = 15 minutes after reaching setpoint
SET1=-55.0	set temp to -55.0°C
WAIT1=2	wait 2 minutes
SET1=125.0	define temp, rate still=10°C/m
NEXT I2	mark end of FOR loop, I2
END	mark end of local prog. and end of STORE
RUN#0	run local prog. 0

## **RUN - TIME OF DAY**

Function: Starts the execution of a local program at a specific time of day.

Syntax: RUN m TIME=hh:mm:ss or RUN #m TIME=hh:mm:ss  
RUN m TIME=NO; to cancel run at time of day

Description: The RUN-TIME command initiates the execution of a local program defined by m, where m selects a local program in the range of 0 to 9, at a specific time of day. Local programs are entered into the controller using the EDIT command from the local keyboard or downloaded from a remote host computer using the STORE command. RUN-TIME execution may be canceled by issuing a RUN n TIME=NO command, where n is any digit 0 to 9. When a RUN-TIME command has been received the PC1000 will blink the front panel LP led to notify the user that it is waiting to run a local program. By using the STATUS command a remote bus controller can determine if the PC1000 is waiting to run a local program. If the PC1000 is turned off, via the front power switch or via an OFF command, The LP led will continue to blink and the PC1000 will turn itself on and run the local program at the specified time of day. The time value is interpreted in 24 hour format.

Restrictions: None

See Also: RUN, EDIT, STOP, STORE, LIST, END

Example:

RUN 0 TIME=17:00:00                      run local prog. 0 at 5:00 PM



## S/!

Function: Transmit and receive ASCII data from the GPIB to the RS-232/422 Serial Interface.

Syntax: !sss...sss transmit string sss...sss to the serial port.  
S receive string from serial interface and send to GPIB port.

Description: The S/! command is a TC01 compatible command which allows the GPIB interface to transmit ASCII data to the RS-232/422 serial interface or receive data from the serial interface. Care should be taken that a serial device does not transmit data to the controller until an S command has been issued, since, data received from a serial device without an S command pending will be interpreted as command data for the PC1000 controller. An S command is required for each string of characters, terminated with a carriage return and/or line feed, to be transferred to the GPIB interface. GPIB lockup time is inhibited for the S command to allow time for a serial device to respond to a prompt message.

Restrictions: GPIB remote interface command only.

See Also: PRINT n, #

Example: S GPIB interface sends request for reply prior to sending a prompt message to insure that the reply message is sent to the GPIB interface by the PC1000.

!send status the string "send status" is sent to the serial interface to prompt serial interface device.

I'm OK serial device response to prompt. PC1000 transfers message "I'm OK" received from serial interface to the GPIB interface.

## SCALE1 and SCALE2

Function: Examine process sensor scale or units of measure

Syntax: SCALE#1?           examine channel 1 scale  
SCALE#2?           examine channel 2 scale

Description: The SCALE command allows a remote interface to determine the scale or units of measure. The type of sensor and scale for each process channel is selected in the CAL menu during or after calibration. The scale is automatically displayed on the front panel LCD display.

Restrictions: Remote interface command only

See Also: CAL Menu

Example: SCALE 1?           examine channel 1 probe scale  
DEG C           controller response, degrees Centigrade  
SCALE 2?           examine channel 2 probe scale  
DEG F           controller response, degrees F

## SET1 and SET2

Function: Set or examine process set point values

Syntax:	SET1=nnn.n	Set channel 1 control point
	SET1=NO[NE]	disable channel 1 control. set=no or none
	SET1?	Examine channel 1 set point. Response will be the set point value or the string NONE if none set
	SET2=nnn.n	Set channel 2 control point
	SET2=NO[NE ]	disable channel 2 control. set=no or none
	SET2?	Examine channel 2 set point. Response will be the set point value or the string NONE if none set

Description: The SET command allows for setting or examining the value at which you wish the PC1000 to control your process to. The set point is one of the three components of a process segment. The process segment consists of the rate of change of the process, defined by the RATE command, from the processes present value to the value defined by the SET command, plus the time specified by the WAIT command to stay at the temperature defined by the SET command. The units assumed by the SET command are those of its sensors scale that are chosen in the probe CAL menu.

To protect your process the PC1000 automatically controls the process set point values as follows. When the PC1000 is turned on the set1 and set2 values are NONE which means that the PC1000 is not to control the process until a valid set is issued. Also when the PC1000 is not running a local program and a STOP command is received the set1 and set2 values will be made equal to NONE. Also after a local program ends its execution the set1 and set2 will be NONE. When the PC1000 is controlling to a set value(s) and power should fail and if the power is off for a length of time longer than the amount that you select in the SDEF menu, for auto power continue, then set1 and set2 will be made equal to NONE.

For a set value to be considered valid by the PC1000 it must be within the range as determined by the currently entered lower limit, LOL, and upper limit, UPL, of its channel. If it is outside of the limit range then a command error is generated and the previous set value remains unchanged. While controlling the process to a set point, should the process value exceed its channels upper or lower limit then the appropriate process control output enable will be disabled thus protecting the process from damage.

Restrictions: none

See Also: RATE, WAIT, LOL, UPL

Example: If SCALE for the channel 1 is in °C

RATE=10.0	set ramping rate for 10.0°C/minute
WAIT=5	set wait = 5 minutes
C1?	read present process value
25.0	controller response, for example 25.0°C
SET1=35.0	set new set point, for example 35°C
SET1?	examine present set temp.
35.0	controller response

If the channel 1 control outputs are enabled then the process will be controlled to 35 deg.

## SDEF MENU

The PC1000 contains default values which may be tailored to your individual system requirements. The values are stored in nonvolatile memory eliminating the need to reload after a power interruption. The default values may be set locally from the control panel by depressing the SDEF and ENTER keys followed by responding to the menu questions on the control panel display.

SDEF           to enter set defaults menu

SET DEFAULTS? (Y/N)           enter yes to set new defaults

DUAL LOOP CTL? (Y/N)  
  CHAN 2 HUMIDITY ?  
    WET DRY HUMIDITY? (Y/N)  
GPIB ADDRESS (0-30):  
GPIB LOCKUP TIME (2-59sec):  
TIMEOUT PRETIME (0-59sec):  
POWER DOWN AUTO RESTART (0-59min):  
D/A OUTPUT A (1=0 to 5,2=-5 to +5):  
D/A OUTPUT B (1=0 to 5,2=-5 to +5):  
D/A OUTPUT C (1=0 to 5,2=-5 to +5):  
D/A OUTPUT D (1=0 to 5,2=-5 to +5):  
SERIAL PORT (1=232, 2=422):  
RS CHAR ECHO ON (Y/N):  
BAUD RATE (1=9600 to 6=300):  
RS DTR/CTS ON (Y/N):  
BUZZER VOLUME (0=OFF,to 3=LOUD):  
LINE FREQ (1=60HZ,2=50HZ):  
CHART CHAN #1 ON A (Y/N)  
  LOW CHART VALUE?  
  HIGH CHART VALUE?  
CHART CHAN #2 ON B (Y/N)  
  LOW CHART VALUE?  
  HIGH CHART VALUE?  
PID#1 TO C D/A? (Y/N)  
PID#2 TO D D/A? (Y/N)  
\*   ENABLE LINE PRINTER  
\*    LINE PER MINUTE 1 TO 6  
\*    LOW CHART VALUE?           {channel 1 charting}  
\*    HIGH CHART VALUE?  
\*    LOW CHART VALUE?           {channel 2 charting}  
\*    HIGH CHART VALUE?  
+/- WAIT #1 TRIG ?  
+/- WAIT #2 TRIG ?

\* = only if optional printer is installed

## SDEF

Function: to read or set PC1000 defaults

Syntax:        SDEF                    to run SDEF menu from front panel  
              SDEF=aaaaaaaa        a=Y,N,1 or 0; n=0 to 3  
              SDEF?

Description: When you wish to setup all PC1000 default parameters then you must enter SDEF from the front panel to enter the set defaults menu. The menu operation is covered under SDEF MENU. When controlling the PC1000 via a remote host computer the SDEF= and SDEF? commands can be used to read and change selected PC1000 default parameters. When the SDEF? command is received the PC1000 sends an 8 character string to the active remote port. To modify the available default parameters the remote host sends SDEF=aaaaaaaa where character position:

1; [Y/N]	Y=dual loop control mode
2; [Y/N]	Y=RS char echo on
3; [Y/N]	Y=chart C1 on D/A A
4; [Y/N]	Y=chart C2 on D/A B
5; [Y/N]	Y=optional printer on
6; [Y/N]	Y=PID #1 to D/A C
7; [Y/N]	Y=PID #2 to D/A D
8; 0 to 3	buzzer volume (0=off)

**Note:** For front panel entry into local programs, a 1 may be used in lieu of a Y and a 0 rather than a N.

Restrictions: Remote command or local front panel menu. The SDEF= command may be stored in LP to modify on the fly.

## SINT MENU

The PC1000 contains interrupt assignment values for local alarm and for the communication ports that may be tailored to suit your system. The values are stored in nonvolatile memory eliminating the need to reload after power interruption. The interrupt values may be set locally from the front panel by depressing the SINT, ENTER keys then responding to the menu prompts on the front panel display.

SINT to enter set interrupts menu

SET INTERRUPTS? (Y/N) ;YES to enter menu

DISABLE ALL INTERRUPTS (Y/N)	;YES to disable
BUZZER LOCAL PROGRAM TIMEOUT (Y/N)?	;YES for buzzer
BUZZER LOCAL PROGRAM DONE (Y/N)?	;YES for buzzer
BUZZER SINGLE TIMEOUT (Y/N)?	;YES for buzzer
INTERRUPT LOCAL PROGRAM TIMEOUT (Y/N)?	;YES for int.
INTERRUPT LOCAL PROGRAM DONE (Y/N)?	;YES for int.
INTERRUPT SINGLE TEMP TIMEOUT (Y/N)?	;YES for int.
DEVIATION INTERRUPT (Y/N)?	;YES for int.
COMMAND ERROR INTERRUPT (Y/N)?	;YES for int.
BKPNT INTERRUPT (Y/N)?	;YES for int.
P_POLL (1-8) 0=NO :	;number of bit

The following interrupt values are sent to the GPIB or RS232/422 ports. The port to send to, if any, is determined by which is currently active. Many of the interrupts may be disabled via the SINT menu. Serial port interrupts consist of ASCII character(s) followed by carriage return and line feed. IEEE-488 (GPIB) interrupt values are the value received when a serial poll is conducted.

### PC1000 INTERRUPT ASSIGNMENTS

Condition	Serial Port <u>ASCII</u>	GPIB Port	
		<u>HEX</u>	<u>DECIMAL</u>
NO INTERRUPT	(N/A)	\$00	0
SINGLE TEMP TIME OUT #1	I	\$41	65
SINGLE TEMP TIME OUT #2	J	\$51	81
COMMAND ERROR INT	CMD ERROR!!	\$42	66
LP TIMEOUT #1	P	\$43	67
LP TIMEOUT #2	Q	\$53	83
LP DONE	E	\$45	69
DEVIATION LIMIT #1	D	\$46	70
DEVIATION LIMIT #2	F	\$56	86
UPL1 EXCEEDED	O	\$47	71
LOL1 EXCEEDED	U	\$4A	74
UPL2 EXCEEDED	+	\$57	87
LOL2 EXCEEDED	-	\$5A	90
GPIB LOCKUP INT	(N/A)	\$4F	79
POWER GOING DOWN	!	(NONE)	
POWER UP NO AUTO CONT.	Z	(NONE)	
POWER UP AUTO CONT.	X	(NONE)	
BKPNT	B	\$50	80

## SINT

Function: to read or control PC1000 interrupts

Syntax:

SINT	to run SDEF menu from front panel
SINT=aaaaaaaaaan	a=Y,N,1 or 0; n=0 to 8
SINT?	

Description: When you wish to setup all PC1000 interrupt parameters then you must enter SINT from the front panel to enter the set interrupts menu. The menu operation is covered in the SINT MENU section. When controlling the PC1000 via a remote host computer the SINT= and SINT? commands can be used to read and change selected PC1000 interrupt control parameters. When the SINT? command is received the PC1000 sends an 11 character string to the active remote port. To modify the available interrupt control parameters the remote host sends SINT=aaaaaaaaaan where character position:

1; [Y/N]	Y=all interrupts off
2; [Y/N]	Y=single timeout int enabled
3; [Y/N]	Y=deviation int's enabled
4; [Y/N]	Y=LP timeout int enabled
5; [Y/N]	Y=LP done int enabled
6; [Y/N]	Y=buzz single time out
7; [Y/N]	Y=buzz LP done
8; [Y/N]	Y=buzz LP time out
9; [Y/N]	Y=cmd error int enabled
10; [Y/N]	Y=BKPNT int enabled
11; 0 to 8	GPIB ppol response bit

**Note:** For front panel entry into local programs, a 1 may be used in leu of a Y and a 0 rather than a N.

Restrictions: Remote command or local front panel menu. The SINT= command may be stored in LP to modify on the fly.

## STATUS?

Function: To poll the PC1000 status

Syntax: STATUS?

Description: The status inquiry command is a remote bus command that allows the host control computer to poll the PC1000 to determine its current operating status. When the STATUS? command is received the PC1000 sends back a 26 character string of Y's or N's where character position:

1; [Y/N]	Y=power on
2; [Y/N]	Y=cmd error last command
3; [Y/N]	Y=time out LED is on
4; [Y/N]	Y=waiting for WAIT1 timeout
5; [Y/N]	Y=C1+ is enables (heat enable)
6; [Y/N]	Y=C1- is enabled (cool enable)
7; [Y/N]	Y=valid SET1 has been entered
8; [Y/N]	Y=waiting for WAIT2 timeout
9; [Y/N]	Y=C2+ is enabled
10; [Y/N]	Y=C2- is enabled
11; [Y/N]	Y=valid SET2 has been entered
12; [Y/N]	Y=DEVL1 exceeded
13; [Y/N]	Y=C1 is ramping
14; [Y/N]	Y=DEVL2 exceeded
15; [Y/N]	Y=C2 is ramping
16; [Y/N]	Y=C1<LOL1
17; [Y/N]	Y=C1>UPL1
18; [Y/N]	Y=C2<LOL2
19; [Y/N]	Y=C2>UPL2
20; [Y/N]	Y=waiting at a BKPNT
21; [Y/N]	Y=in LP run mode
22; [Y/N]	Y=in LP remote store mode
23; [Y/N]	Y=in local LP edit mode
24; [Y/N]	Y=waiting to run LP at TOD
25; [Y/N]	Y=GPIB bus timeout
26; [Y/N]	Y=in local lockout

Restrictions: Remote only, read only, not stored in LP



## STOP

Function: Terminate local program execution and single temperature segment execution.

Syntax: STOP

Description: If in Local Program operation, the STOP command terminates local program execution. If running in single mode, then entering the STOP command sets SET1=NONE, SET2=NONE and WAIT1=FOREVER, and WAIT2=FOREVER.

Restrictions: None

See Also: RUN

Example:

DELP#0	clear LP area #0
STORE#0	ready controller for LP commands
***	send LP commands
END	terminate store operation
LIST#0	display local program
***	controller displays local program
RUN#0	execute local program
***	
STOP	terminate local program activity

## STOPE9

Function: To do a reset of PC1000 software.

Syntax: STOPE9                      STOP key, E key, 9 key, enter

Description: The STOPE9 command will restore SDEF and SINT parameters to factory default settings, clear all local program data, and initialize all internal PC1000 software variables. This command does not modify the sensor calibration data that was generated in the CAL menu. This command is provided as a software reboot. The PC1000 will go to the power off mode after receiving this command. When you turn the PC1000 back on the first time a ram error will normally be displayed.

**Note:** All local program data will be erased. You may wish to upload local programs to a remote computer before issuing this command if you have large programs that you wish to save.

## STORE

Function: Stores a local program from a remote interface

Syntax: STORE m or STORE #m

Description: The STORE command allows a remote interface to download a local program, where "m" identifies one of ten possible local program areas, 0 thru 9. Once downloaded, local programs may be executed by issuing a RUN command. Prior to downloading a program to the controller, the local program must be cleared by using the DELP command. Local programs can be uploaded to a remote computer by using the LIST command. When the PC1000 receives the STORE command it automatically sends, to the active remote port, the number of free local program bytes available so that the sender can determine if the program it wishes to send will fit into the PC1000 memory. To calculate the size of the program, simply count the number of characters that you are sending plus one byte for each line.

Restrictions: Remote interface command only.

See Also: LIST, RUN, DELP, STOP

Example:

```
DELP#0      clear LP area #0
STORE#0     ready PC1000 for accepting LP commands
1245       PC1000 sends free byte count
***        send LP commands
END         terminate store operation
LIST#0list  local program to remote
***        PC1000 sends local program
RUN#0      execute local program
***
```

## T

Function: Examine current channel 1 temperature in °C

Syntax: T

Description: The T command provides upward compatibility to one of SUN's older controllers. The T command is a TC01 controller compatible command which examines the present temperature of channel 1 probe. The response is always in degrees centigrade. For new applications the C1? command should be used.

Restrictions: Remote interface command only.

See Also: C1?

## TIME

Function: Set or examine battery backed time of day clock

Syntax:

TIME=hh:mm:ss	set time of day
TIME?	examine present time of day

Description: The PC1000 has a built in battery backed time of day clock that is used by the PC1000 to determine how long line power is off, during a power line blackout, and when it is time to automatically run local programs. The TIME= command is provided so that the internal clock may be set to your local time while the TIME? command is provided for documenting report outputs, etc. Time is entered and displayed in a 24 hour format, i.e., hours (hh) range from 00 to 23; minutes (mm) range from 00 to 59 and; seconds (ss) range from 00 to 59.

Restrictions: None

See Also: RUN #m TIME=hh:mm:ss

Example:	TIME=13:30:00	set time to 1:30 PM
	TIME?	examine time of day
	13:30:02	controller response

## TIMEE?

Function: To determine number of hours of use

Syntax: TIMEE?

Description: The PC1000 keeps track of the number of hours that the unit has been on. The time is saved in EEROM to protect against data loss. The TIMEE value is updated whenever the PC1000 is turned off or when at least one hour has elapsed since the last update.

Example:	TIMEE?	ask for number of hours of use
	27.25	controller response

## UPPER LIMIT UPL1 & UPL2

Function: Set or examine the upper temperature limit

Syntax: UPL1=nnn.n Set high limit for channel 1  
UPL1? Examine high limit channel 1  
UPL2=nnn.n Set high limit for channel 2  
UPL2? Examine high limit channel 2

Description: The UPL command allows for setting the highest process set value that the PC1000 will accept as a valid setting. If the process should ever goes above the UPL value, the heat or increase enable for that channel will be disabled and an interrupt to the active remote will be generated. The assumed units for the UPL command are those as selected for its channel, as set in the CAL menu.

Restrictions: The range of values for the UPL commands are determined by the type of sensor selected for its channel.

See Also: LOL1, LOL2

Example: assume RTD sensor channel 1 °C as scale  
UPL=100.0 set UPL to 100°C  
UPL? examine UPL  
100.0 controller response, 100°C

## VER?

To remotely determine the version number of the firmware that is installed in your PC1000, the VER? command may be used. The firmware version is also displayed at power on.

Example: VER? Host computer sends version command  
SUN SYSTEMS 3.10 version message returned

## WAIT1 & WAIT2

Function: Set or examine the process segment wait period

Syntax:

WAIT1=hh:mm:ss	set wait 1 in hours:minutes:seconds
WAIT1=mm	set wait 1 in minutes (mm=0 to 59)
WAIT1=F(OREVER)	set WAIT 1 to forever
WAIT1?	examine current wait 1 setting
WAIT2=hh:mm:ss	set wait 2 in hours:minutes:seconds
WAIT2=mm	set wait 2 in minutes (mm=0 to 59)
WAIT2=F(OREVER)	set WAIT 2 to forever
WAIT2?	examine current wait 2 setting

Description: For most process control applications it is necessary to control the process to a set point and after reaching the set point stay at that set point for some period of time. The WAIT command determines the length of time that the controller will maintain the process, after the process is within the +/- trigger window. The +/- trigger window value is specified in the SDEF menu. Once the wait timing begins, the wait time continues to count down regardless of changes in the measured process. The WAIT command is one of the three commands that may be used to define a process control segment. RATE controls the rate of change, WAIT determines how long to stay at the set point, while SET enters the target set point. It is recommended that the three commands be entered in the before mentioned order to insure that before the set point is entered, the PC1000 knows at what rate to move and how long to stay there.

When entering a RATE-WAIT-SET segment in the non local program mode, when the wait time is over, the PC1000 generates a timeout interrupt/buzzer and continues to hold the process. When running a local program, the next command in the program is executed after the wait period has timed out. During the wait period, examination of WAIT returns the time remaining in the wait period. After the end of a wait timeout, the WAIT is reset to forever.

Restrictions: Range from 99:59:59 to 00:00:01

See Also: SET, RATE,

Example:	define temperature segment
	RATE1=10 , rate=10°C/min
	WAIT1=00:10:30 wait=10 min, 30 sec
	C1? examine current process reading
	25.0 controller response, 25.0°C
	SET1=35.0 set new temperature, 35°C
	after 1 minute, the ramp is done
	wait period begins
	after 1 more minute,
	WAIT1? examine wait setting
	00:09:30 controller response, 9 min, 30 sec remaining

## #TEXT

Function: send ascii string sss to printer

Syntax: #sss...sss

Description: This command allows you to send an ascii string from a remote host to the optional line printer. If the printer option is not installed then the command is ignored. This command provides for user documentation of PC1000 line printer report output.

Example: #PRINT THIS STRING

?

The ? command is used to get status of last command that was given to the PC1000. You always get a 2 line response

If previous command caused an error then the offending command line and a pointer to the first offending symbol are returned. Example: you send RATT1=27 command, a command error is generated if enabled in SINT menu. To determine reason for error you send ? command. The 2 line response would be

```
RATT1=27 <<
  ^
```

The ^ character points to the point where syntax was found to be incorrect (expected E got T). If the reason for the command error was not syntax related then an explanation for the error is returned in line 2. For example: if the PC1000 is set for single loop control, via SDEF menu, and you issue a SET2=63.4 command the 2 line response to the ? command would be

```
SET2=63.4
NOT IN DUAL MODE
```

If the last command was good then the PC1000 will send back a two line OK, OK response. For example if you send WAIT1=3 then send the ? command, the 2 line response will be;

```
OK
OK
```

The ? command can also be used to remotely determine how the PC1000 has powered up. After the PC1000 is turned on, or after the unit powers up from a power failure. The ? command must be the first command issued to get the power up status information. For example you turn the PC1000 on via a ON command, then send the ? command, the two line response will be;

```
SUN SYSTEMS 3.10
SELF TEST OK
```

or if there was a problem during the power on self test a 2 line message similar to the following will be returned;

```
SUN SYSTEMS 3.10
RAM ERROR
```

## VI PID COEFFICIENT TUNING AND ADVANCED PID MODE CONTROL

SUN ELECTRONIC SYSTEMS, Inc.

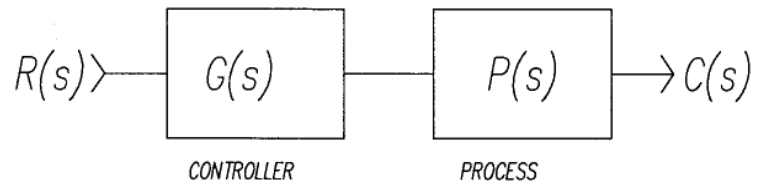
### ENGINEERING NOTE

#### PURPOSE:

To provide the user with detailed information for optimum use of the built in PID control loop features. It is assumed that the PC1000 is controlling an air temperature chamber to illustrate the PID features.

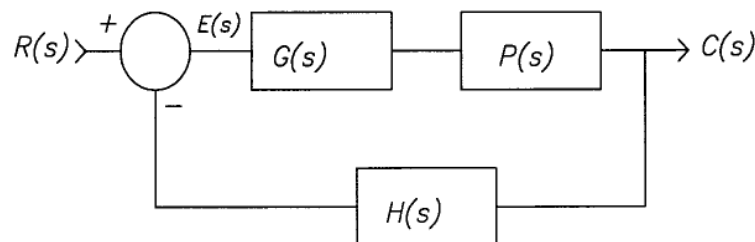
#### DEFINING FEEDBACK CONTROL:

Control systems can be classified as "open loop" or "closed loop". Shown is the block diagram of an open loop system. In this type of system the set or desired process value,  $R(s)$ , is processed by the controller and then drives the process



to be controlled. The open loop type of control is fine so long as the transfer function of the control,  $G(s)$ , and the process,  $P(s)$ , remain constant. In the real world that is seldom the case. For example the controller may be an analog amplifier whose gain varies with supply voltage and ambient temperature. If the gain were to change by 10% then the process value would also change by a like amount. The process will also most probably change due to some variable, for example the dynamics of a chamber will change when the user places his product, to be temperature tested, in the chamber. The open loop system does have a couple of advantages. First is simplicity and the second is that it is always stable.

To automatically compensate for the inevitable changes in the controller and process the closed loop control feedback loop was devised as shown in following Figure. The prime difference between the open and closed loop system is the generation and utilization of the error signal  $E(s)$ , which is the difference between the desired and measured process value. By using the error signal for control, sensitivity to controller or process variations is significantly reduced. To illustrate the improvement, let us consider a change in the transfer function  $G(s)$  such that we have  $G(s)+\delta G(s)$ . Then in the open loop case, the change in the transform of the output is:



$$C(s) = G(s)R(s)$$

or  $C(s)/R(s) = G(s)$

To find change in C(s) with respect to change of G(s)

we have:  $dC(s)/R(s) = G(s)'$

or the process change will follow the controller change.

In the closed loop system we have:

$$C(s) = G(s)R(s) / ( 1+G(s)H(s) )$$

or  $C(s)/R(s) = G(s) / ( 1+G(s)H(s) )$

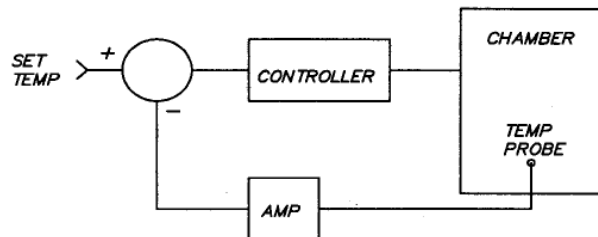
To find change differentiate thus:

$$\frac{d(C(s)/R(s))}{dG(s)} = \frac{( 1+G(s)H(s) )G(s)' - G(s)G(s)'H(s) }{( 1+G(s)H(s) )^2}$$

or  $\frac{d(C(s)/R(s))}{dG(s)} = \frac{G(s)'}{( 1+G(s)H(s) )^2}$

or the change in the transfer function of the closed loop system is reduced by a factor  $(1+G(s)H(s))^2$  which is designed to be much greater than 1, thus greatly reducing process variation.

In a closed loop temperature control system, as shown in figure, a temperature sensor measures the chamber temperature. The amplified feedback signal from the sensor is compared with the desired set temperature to produce the error temperature. If the error is not zero then the controller adds or removes heat to the chamber in an attempt to force the error to zero. A closed loop system thus operates to reduce the error signal to zero so that the process temperature and the set temperature are equal. Just how the control section accomplishes this task is the subject of this paper.



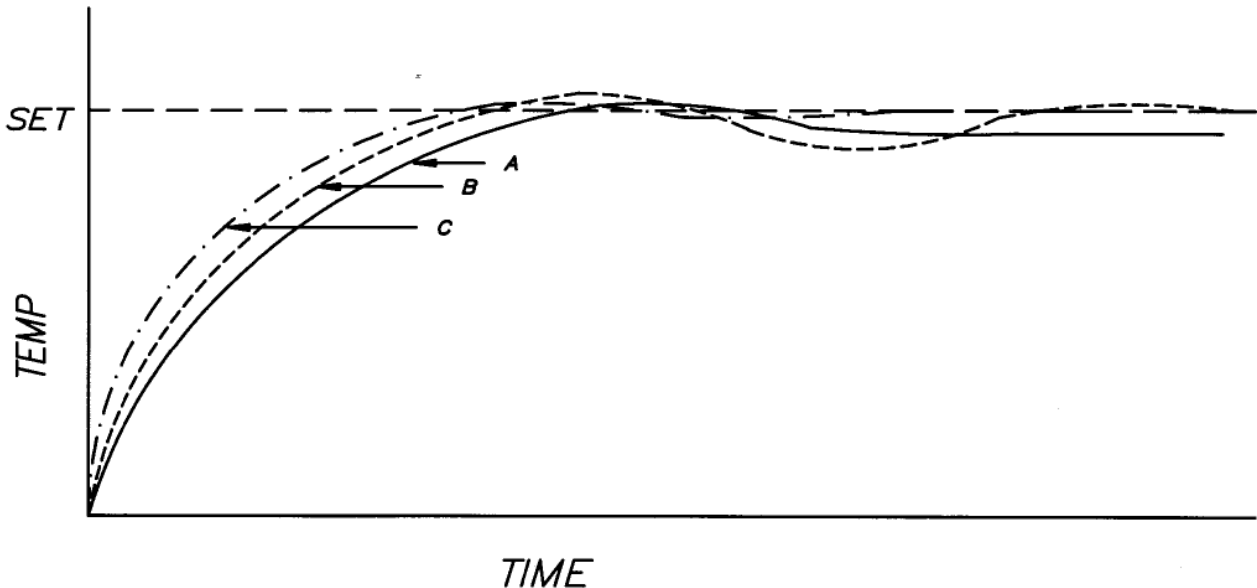
### CONTROL SYSTEM RESPONSE

The manner in which the system responds to a new set point, or an external disturbance, is referred to as the control system response. The two most common disturbances used to test a control systems response are a step change (A/S) or a ramping change (A/S\*\*2) of the set point. Depending upon the application, a variety of system responses can be designed by changing the forward loop controller coefficients. Our Microprocessor based digital control systems offer the flexibility of adjusting the controller response through easy to use software commands, providing the capability of tailoring the system precisely to a given application, should it be necessary.

Since 1978 when we introduced our first digital process controller we have used the PID as the control function of choice. PID is an acronym for Proportional, Integral, and Derivative.



Proportional function is the most basic controller type and is a component of almost all control systems. In a proportional only system, the corrective action is proportionate to the difference between the desired set point and the feedback from the process. In other words, as the error becomes smaller the corrective action reduces. Figure curve A shows the response of a proportional only control system to a step change of the set point. Note that the proportional response results in an offset or continuous error since the controller output approaches zero as the error approaches zero. This requires that there be some error signal to provide an output to maintain the process. The offset error can be reduced by increasing the proportional gain however there is usually an upper limit to the loop gain that will still provide a stable system.



In many instances the inevitable offset error, indicative of a proportional gain only controller, cannot be tolerated. One way to eliminate the error is to modify the controller response so it also responds to how long the error is present. This is done by integrating the error signal over time and combined with proportional gain results in an overall system response as illustrated in figure curve B. As shown the addition of the integral feature removes the offset error. At this point we have a PI controller.

A further reduction in settling time and overshoot, or increased stability, can be obtained by also designing the controller to respond to how fast the error is changing. A response to how fast the error is changing is called a rate or derivative response. The response of a controller that combines the Proportional, Integral, and Derivative features is illustrated in figure curve C. This is the full PID controller response. As shown, a properly tuned PID system provides the shortest settling time and minimum overshoot of the three responses shown. The PC1000 uses the full PID control algorithm, and allows the user to set the P, I, and D weighting coefficients should optimization be required. Thus the user may "TUNE" the PID coefficients for an optimum control system response. The PC1000 provides for two sets of PID coefficients for each channel. One set is used when the controller determines that cooling is required and another set is used when heating is needed. Two PID sets are provided since occasionally the process response during cooling may be different than during heating.

The questions to be answered now are "when do I need to change the default PID coefficients and how do I determine new coefficients". We at SUN have tried to design the PID control algorithms to be as robust as possible so that tuning of the PID coefficients is often unnecessary. The controller even does slight self tuning of its coefficients to automatically adapt itself. However should the user wish to change the response to his particular needs, the ability to change the coefficients is provided. To address the question of determining new coefficients we next provide a simple iterative approach.

## **ITERATIVE PID COEFFICIENT DETERMINATION:**

Since mathematically modeling the control system requires that the model coefficients of the process and sensor be determined, which in most circumstances is an undesirable use of time, the following iterative approach is presented. In the following the step input is performed by setting RATE=500 to allow for fast rate of change and to give the PC1000 a new set point that is significantly different than its current process value. The time response is determined by monitoring the process value. The optimization can be performed manually or with the aid of a remote bus controller by plotting the process versus time after the step is given. If you have the optional line printer then it will automatically provide the response printout. Another point that must be kept in mind is that some applications provide different amounts of energy to the process dependent upon whether it is heating or cooling. Now tuning one channel at a time;

### **START WITH THE HEATING PID COEFFICIENT SET ( PID1+ )**

1. Start by finding a proportional coefficient that provides a stable response to a step change of the set point. The Integral and Derivative coefficients should be set equal to 0.0 during this phase. You should get a response similar to figure curve A, looking for small or no overshoot. Note that the steady state error should be ignored for the present. As a starting point the default P coefficient in the PC1000 when shipped is 0.25. Also note that  $1.0/P$  coefficient equals the error band. In other words the 0.25 default coefficient provides a 4.0 degree C error band so as soon as the error becomes less than 4.0 deg C the controller begins to reduce the output from full on.
2. Next bring up the Integral coefficient until the steady state error is reduced to an acceptable value in a reasonable time period. If the system becomes unstable by increasing the Integral coefficient, then reduce the Proportional coefficient. As a starting point the default I coefficient is 0.001 as shipped. Note that the D coefficient is equal to 0.0 during this phase.
3. Finally bring up the derivative coefficient to help reduce any overshoot and to increase loop stability. As a starting point the default D value is 0.10.

IF YOU WISH TO TUNE COOLING, REPEAT STEPS 1,2, AND 3 FOR COOLING COEFFICIENTS (PID1-), OR USE THE VALUES THAT YOU FOUND FOR THE HEATING PID'S. Then tune channel 2 if required.

### **ADVANCED PID MODES OF OPERATION (PIDA)**

To illustrate the use of the advanced PID modes, we will assume that the PC1000 is connected to an air temperature chamber. The PC1000 is in single channel control. The two sensor inputs will be connected to temperature probes. The two probes are referred to as the chamber air probe and the user probe. The chamber probe is located at the top of the chamber to sense air temperature. The user probe is available in the chamber workspace to be placed as the user desires. The user probe is provided so that the temperature of a users Device Under Test (DUT) can be determined or, through the use of the advanced PIDA modes, be controlled. The PC1000, for the first time, allows the test engineer to automatically control the test sample temperature while still providing stable and predictable chamber air temperature control. To illustrate the utility of the dual probe chamber, let us discuss how things are currently done. Assume you need to test your DUT at -55 deg C and at 155 deg C. There are two common techniques used to accomplish this type of test.

The chamber air temperature would be set to a set temperature. Then the temperature of the DUT would be monitored by another temperature meter to determine the length of time it takes the DUT to reach the test temperature. This time would then be added to the soak time. For a DUT with large thermal mass the time will be considerable. To shorten the test time the user may intentionally overshoot the test temperature then revert back to the actual set point. Not only do you need an extra temperature meter for setup, but since the DUT is not really in the control loop this technique will not compensate for changes in DUT size, testing temperatures etc.

Another approach is to place the chamber temperature sensor on the DUT so that its temperature is being controlled rather than the air temperature. The disadvantage of this approach is that the DUT will add a thermal lag or delay such that the chamber air temperature will overshoot the set point by an amount that may damage the smaller parts of the DUT or cause chamber problems. The thermal lag caused by the DUT may also cause the chamber control to become unstable.

To address the above described problem SUN SYSTEMS provides a number of advanced control modes that the user can select from to simplify his temperature testing. The PC1000 advanced PID mode is controlled using the following:

PIDA?            To determine current PID mode

PIDA=n[,m]      To place controller in a new mode

WHERE:          n=0 or 1 or 2 or 3 or 4  
                  m=optional damping ratio when n=2 or 4

n=0              Normal PID mode ( control CH1 to SET1 )

n=1              Average mode (control to {CH1 + CH2}/2)

n=2              Slowly force CH2 to set1

n=3              Use CH2 probe as control probe

n=4              Control to average and slowly force CH2 to set1 point

WHEN: n=2 or n=4 the optional damping can be set. The default value is 400.0. The larger the damping number the slower the controller will move the CH2 probe to the set1 point. Through the use of this damping coefficient the user can control the air temperature overshoot.

OTHER: The PIDA modes also change how the controller determines when it will start wait timing, as follows.

PIDA=0          when  $ABS( set1 - CH1 ) < trigger\ 1$

PIDA=1          when  $ABS( set1 - (CH1 + CH2)/2 ) < trigger\ 1$

PIDA=2          when  $ABS( set1 - CH2 ) < trigger\ 1$

PIDA=3          when  $ABS( set1 - CH2 ) < trigger\ 1$

PIDA=4          when  $ABS( set1 - CH2 ) < trigger\ 1$

where trigger 1 is the value set up in SDEF menu

The following six printouts give example temperature verses time charts using the advanced

PID modes. In all of the examples the user probe was attached to an 11 ounce aluminum block DUT laying on the inner chamber floor.

Example number 1 is provided as a mode 0 base line. It shows how long it takes to get the DUT to the chamber air temperature in a standard chamber air temperature controlled configuration. After 40 minutes the DUT is still 5 deg or so from the set temperature.

Example number 2 demonstrates PIDA=1 where the average of the chamber probe and user probe temperature is controlled to the set point. As shown the DUT is within about 2 degrees of the set temperature after 30 minutes. The maximum chamber air temperature overshoot was about 28 deg C. Since both probes are used to generate the error signal the control loop stability is good.

Example 3 gives the time response of PIDA=2 using a default damping of 400. As shown the DUT gets to the set point within about 15 minutes. There is approximately 21 deg of air temperature overshoot and some DUT overshoot.

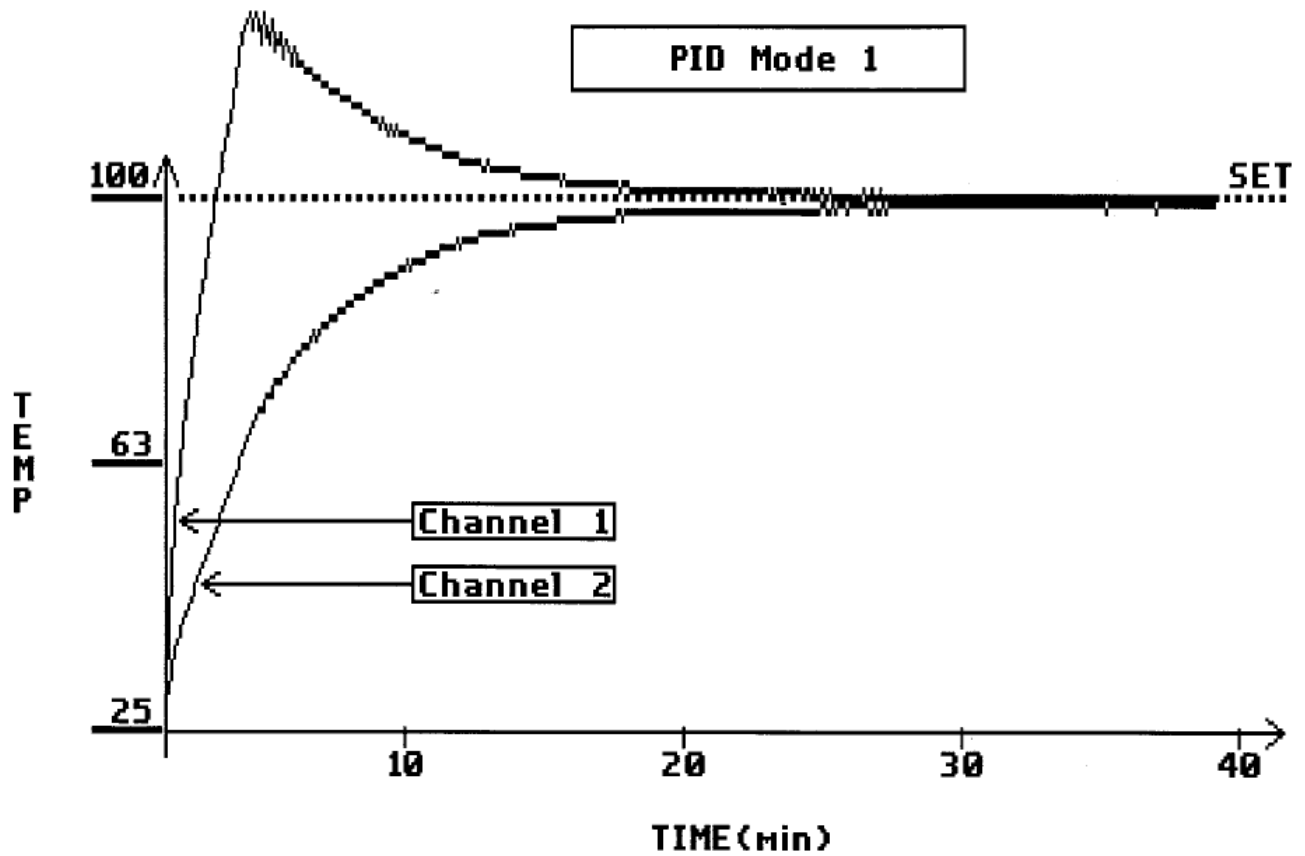
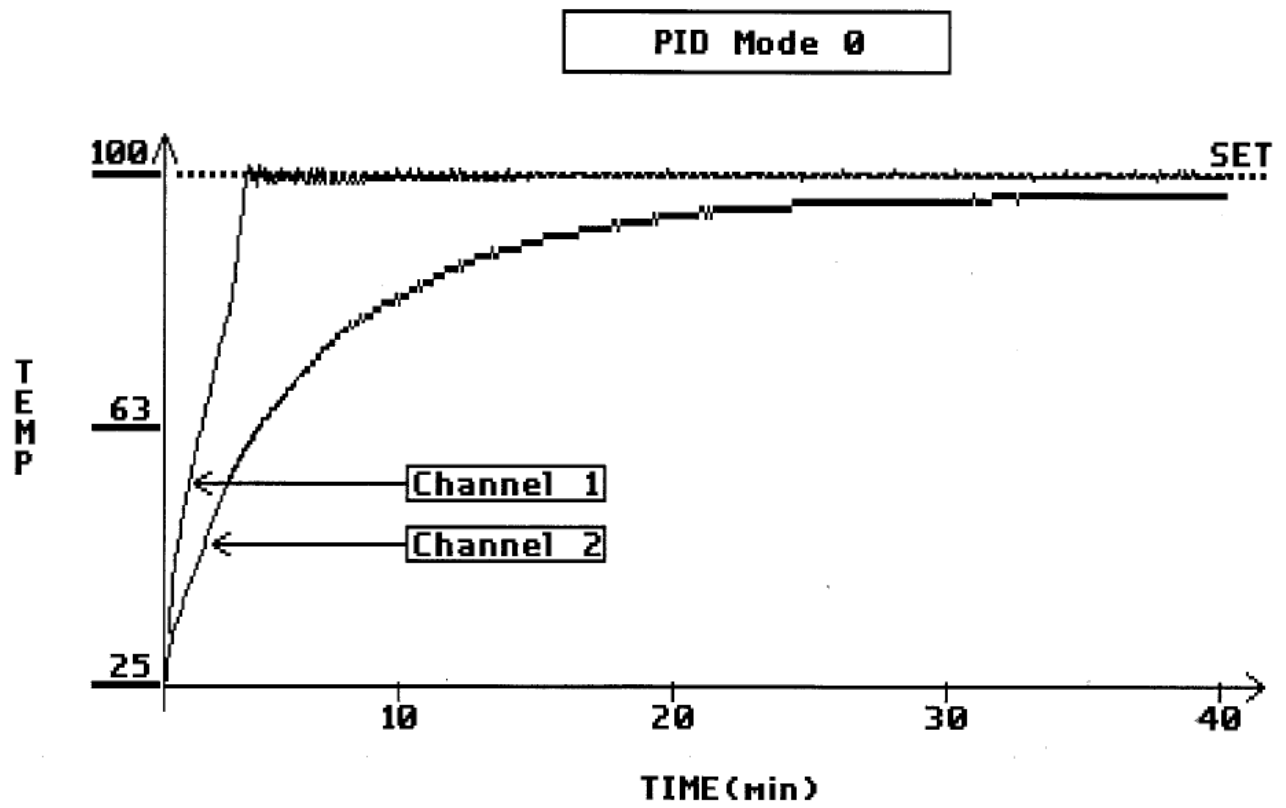
Example 4 shows how the system behaves when PIDA mode 3, or controlling to the user probe is selected. When the DUT is not very large, as in this case, and when large air temperature overshoot can be tolerated this mode is quite useful.

Finally example 5 demonstrates PIDA=4,400 or controlling to the average of chamber and user probe while slowly forcing the user probe to the set temperature. A reasonable air overshoot is shown with fast DUT to set temperature.

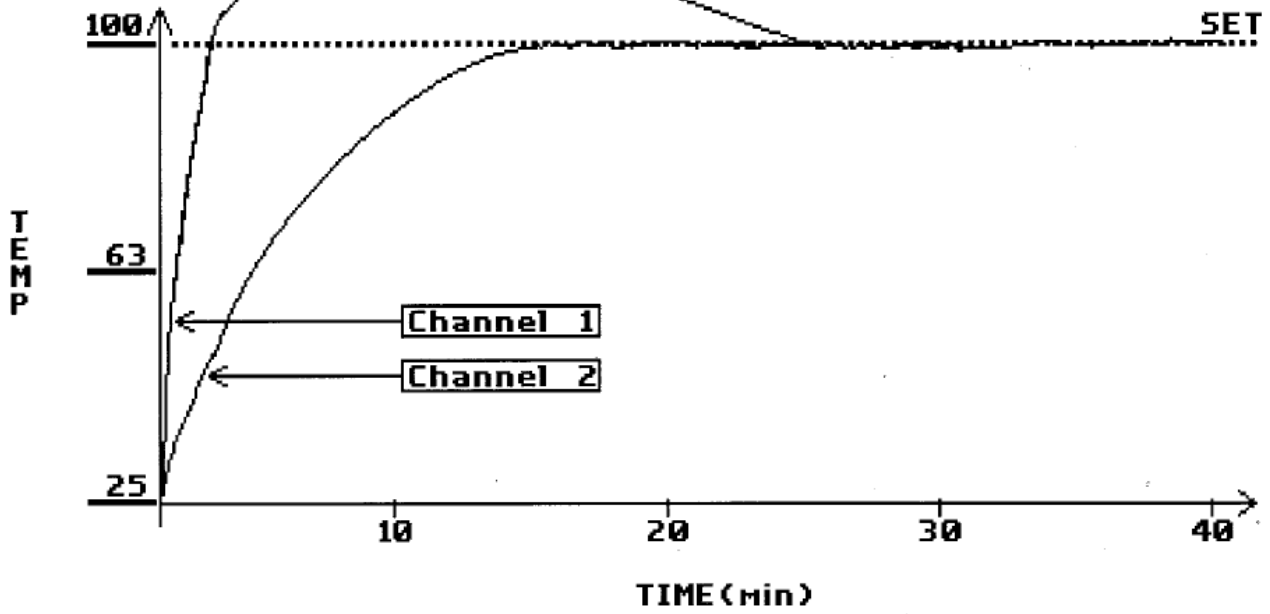
Selection of which advanced PID mode to use should be determined by the user based on allowable chamber air temperature overshoot, the amount of time available to bring the DUT to the set point, and thermal mass of the DUT.

The following is a local program that can be used to produce example 5 printout. An optional printer is assumed to be installed in the PC1000.

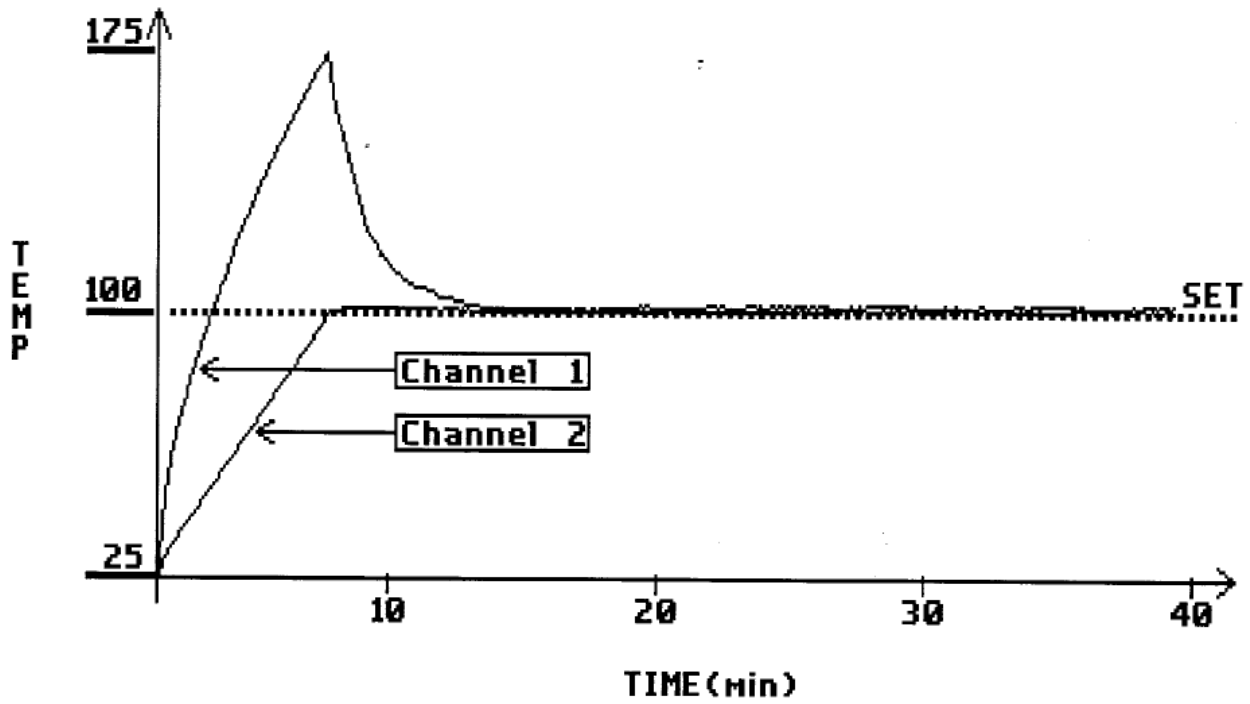
```
SDEF=YNNNNNN3      { TURN OFF PRINTER }
RATE1=100           { SET RATE TO LARGE NUMBER }
PIDA=0              { NORMAL AIR TEMP CONTROL }
WAIT1=30            { GO TO 25 deg FOR 30 MINUTES }
SET1=25
SDEF=YNNYNNN3      { ENABLE LINE PRINTER }
WAIT1=40
PIDA=4,400          { SET ADVANCED PID MODE }
SET1=100            { DO TEMP STEP TO TEST PIDA }
END
```

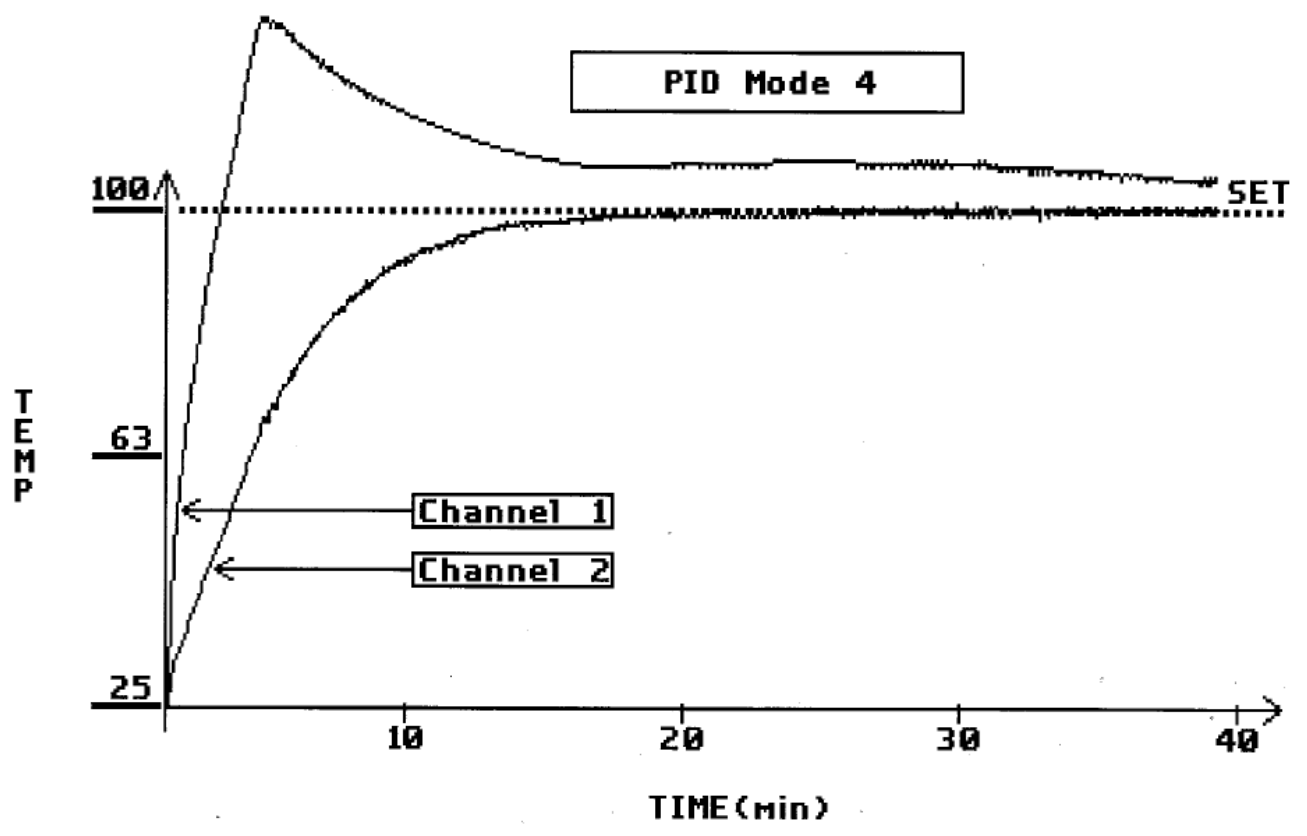


PID Mode 2



PID Mode 3









## **VII PC1000 RELATIVE HUMIDITY CALCULATIONS**

The PC1000 will automatically calculate relative humidity for channel 2 when relative humidity mode has been selected in the SDEF menu. You can chose between dry/wet mode, where the dry bulb sensor is channel 1 and the wet bulb is on channel 2, or dry/dew mode, where the dry is channel 1 and the dew point sensor is channel 2. The PC1000 automatically performs the following;

To calculate RELATIVE HUMIDITY in percent from dry/wet or dry/dew temperature measurements.

For DRY/WET bulb:

$$\text{PERCENT RELATIVE HUMIDITY} = (E / \text{emd}) * 100$$

For DRY/DEW bulb:

$$\text{PERCENT RELATIVE HUMIDITY} = (\text{emw} / \text{emd}) * 100$$

$$\text{where } E = \text{emw} - .00066 * (1 + .00115 \text{ tw}) * P * (\text{td} - \text{tw})$$

and:

E = VAPOR PRESSURE

emw = SATURATION VAPOR PRESSURE FOR WET BULB TEMP

emd = SATURATION VAPOR PRESSURE FOR DRY BULB TEMP

tw = TEMPERATURE OF WET BULB IN C

P = ATMOSPHERIC PRESSURE IN mb ( 1 atm=1013.246 mb)

td = TEMPERATURE OF DRY BULB IN C

The two saturation vapor pressures, emd and emw, are found in the Smithsonian Meteorological Tables. Using these tables a set of least square fit coefficients were calculated. emd and emw are both found in the same table. Two tables exist, though. One for Saturation vapor pressure over water and another for over ice. The pressure is also different over ice relative to over water. In the case of DRY/WET bulb humidity measurement, the dry bulb temperature is used during calculation of emd and the wet bulb temperature is used to calculate emw. In the case of DRY/DEW bulb humidity measurement, the dry bulb temperature is used during calculation of emd and the dew point temperature is used to calculate emw.



## VIII PRINTER OPTION

The printer option is a printed circuit board that is installed in the PC1000, if ordered at the time of purchase, or is a module that plugs into the USER port, if ordered separately. The function of the printer option is to provide a IBM Personal Computer type printer port for PC1000 data logging and to provide for an extra 8 bits of input and output. The printer port is designed to use a standard PC parallel printer cable and an inexpensive dot matrix line printer, such as a Panasonic PanaPrinter. The bit I/O is provided on a 26 pin flat cable port with the same pinouts as the standard PC1000 process port, except that another fail-safe input is not provided. The same process bit I/O printed circuit board is available for screw terminal connection to the extra I/O bits. To access the extra I/O use the standard OUT and IN commands where the bit addresses are 8 to 15.

To use the printer for automatic data logging, first unplug the PC1000 then connect the printer and then plug the PC1000 back in. The PC1000 will look for the printer option hardware during power up. It will first look for the internal printer hardware then if not there will look on the user port. Then to setup the auto data logging you run the SDEF MENU. If the printer hardware was found the menu will ask:

ENABLE LINE PRINTER	answer yes
LINE PER MINUTE?	answer 1 to 6
LOW CHART VALUE?	chan #1 low chart value
HIGH CHART VALUE?	chan #1 high chart value
LOW CHART VALUE?	chan #2 low chart value
HIGH CHART VALUE?	chan #2 high chart value

Now whenever there is a valid set1 or set2 the PC1000 will produce an output plot.



## **IX FIRMWARE UPGRADE INFORMATION**

The firmware version is displayed when the PC1000 is turned on or through the use of the VER? command. We at SUN are constantly working on improving the capabilities of our products. If you wish to obtain new PC1000 firmware as it becomes available, please call us with your current version number. The firmware upgrade consists of the EPROM that contains the PC1000 control program and documentation of new features. The cost of the upgrade is \$100.

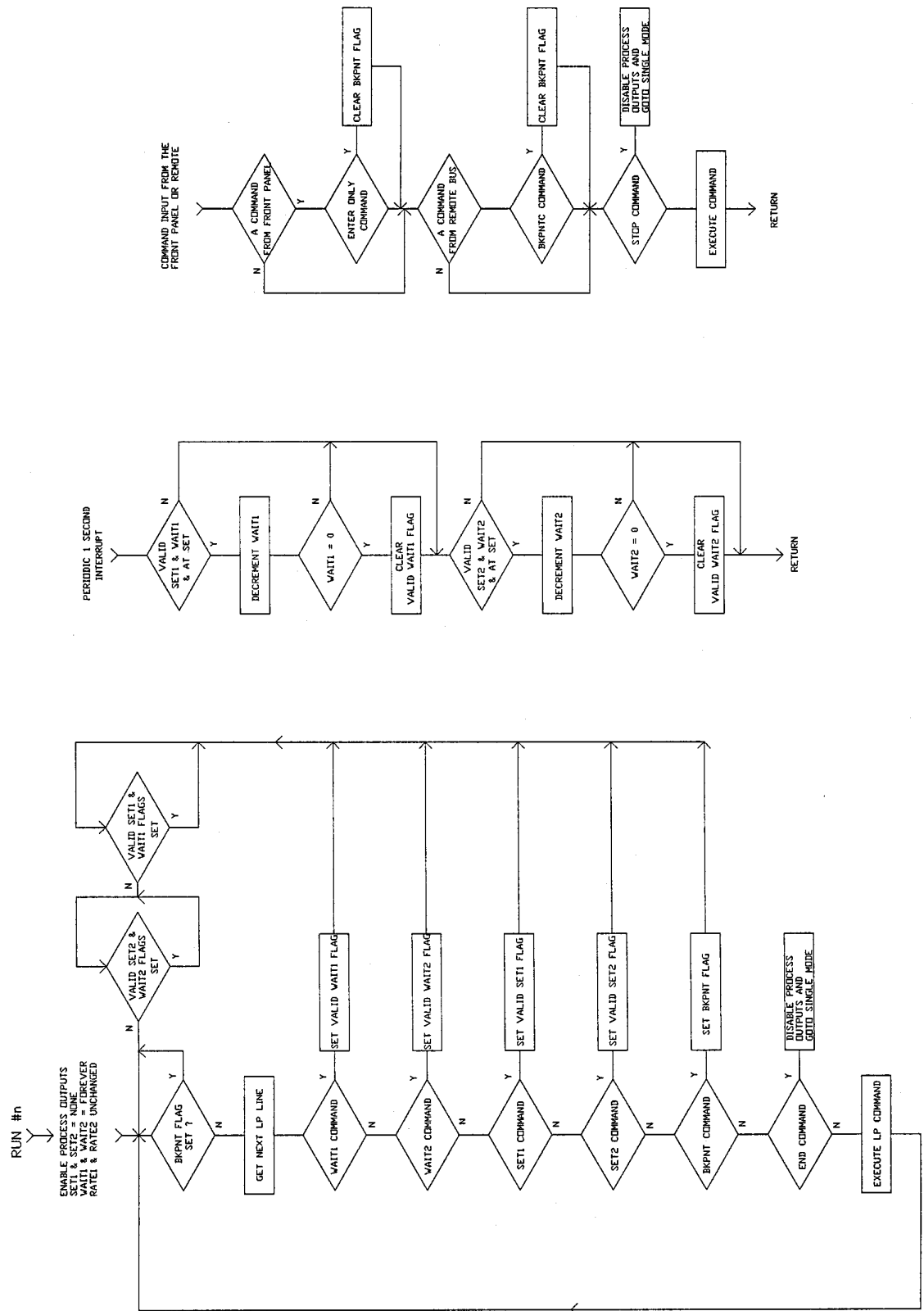
### FIRMWARE UPGRADE INSTRUCTIONS (NEW EPROM INSTALLATION)

1. YOUR LOCAL PROGRAMS AND DEFAULT SETTINGS WILL BE LOST WHEN THE NEW EPROM IS INSTALLED. PLEASE WRITE DOWN YOUR SPECIAL SETUP INFORMATION AND LOCAL PROGRAM INFORMATION BEFORE PROCEEDING. YOU MAY WISH TO UPLOAD LOCAL PROGRAMS TO A PC DISK FILE USING THE RS232 INTERFACE SOFTWARE PROVIDED on the floppy disk.
2. Unplug PC1000 then remove cover. Find the socketed EEPROM which is U1 at about the center of the main logic circuit board. Remove U1 and replace with new EEPROM. Be careful to orient dot or notch on chip correctly.
3. Replace PC1000 cover and plug in unit.
4. FROM FRONT PANEL TYPE IN THE FOLLOWING COMMANDs:  
  
          STOPE9      STOP KEY,E KEY,9 KEY THEN ENTER
5. Turn the PC1000 on then enter;  
  
          CALC,9      CAL, C, comma, 9 then ENTER
6. RUN SDEF, SINT AND RELOAD LOCAL PROGRAMS AS REQUIRED.



# X PC1000 LOCAL PROGRAM FUNCTIONAL FLOW CHART

PC1000 LOCAL PROGRAM FUNCTIONAL FLOW CHART







## **XI PC1000 SPECIFICATIONS**

### **POWER REQUIREMENTS**

POWER CONSUMPTION	35 watts
INPUT VOLTAGE	110/220 +/- 10%
	Selectable at rear panel power input
LINE FREQ	50 or 60 HZ

### **MECHANICAL**

PACKAGE FORM	3.5" high 19" rack mountable or 3.5" by 16.5" bench top
FRONT PANEL	32 key 32 char LCD
REAR PANEL	contains all I/O ports

### **ENVIRONMENTAL**

AMBIENT TEMPERATURE	5 to 45 deg C
	local junction compensation provides .05 deg C/deg C

### **SAFETY**

SENSOR MALFUNCTION	open/short probe detect
CONTROLLER MALFUNCTION	hardware watch dog
PROCESS ERRORS	external fail safe in
PROCESS LIMITS	upper, lower and process deviation limits
LINE VOLTAGE LOSS	battery backed memory and time of day clock. setable auto restart after power loss

### **PROCESS SENSOR INPUT CHANNELS**

RANGES:	
RTD (.385 or .392)	-200 to +320 deg C
J narrow	-200 to +320 deg C
K narrow	-200 to +320 deg C
T	-200 to +320 deg C
J wide	-200 to +760 deg C
K wide	-200 to +1250 deg C
SOLID STATE	-60 to +160 deg C
VOLTAGE	0 to +5 volts
CURRENT	0 to 20mA
RESOLUTION	15 bit conversion .02 deg C for RTD and narrow therm .15mV for volts, .61 uA for current

### **ABSOLUTE ERROR OVER RANGE (Note 1)**

RTD's	+/- 0.2 deg C
J & K narrow	+/- 0.35 deg C

J & K wide	+/- 0.75 deg C
VOLTAGE	+/- 500 uV
CURRENT	+/- 2 uA
ELECTRICAL ISOLATION	240 VOLT

## **OTHER FEATURES**

Menu selection of scale ( C, F, K, %RH, USER )  
Menu selection of probe type and range.  
Local reference junction compensation.  
Auto zero and auto gain compensation.  
Software digital filtering.  
Auto integrate over power line period  
Open/short probe detection.  
Menu selection and cal of process sensor.

## **IEEE-488 BUS INTERFACE (GPIB)**

APPLICATION	remote control
PROTOCOL	talker/listener/spol/ppol
BUS ADDRESS	0-30 set in menu
INTERRUPT CAPABILITY	SRQ and PPOLL

## **RS232/422 SERIAL INTERFACE**

APPLICATION	remote control
PROTOCOL	ASCII char line oriented
DATA RATES	300 to 9600 baud, menu selectable

## **4 CHANNEL, 8 BIT A/D INPUTS**

INPUT RANGE	0 to 5 V, 0 to 20mA on first channel
ACCURACY	+/- 1 LSB
CONVERSION TIME	26uS

## **4 CHANNEL, 8 BIT D/A OUTPUTS**

OUTPUT RANGE	menu selectable 0 to +5 V or -5 to +5 with 0 to 20mA one channel
ACCURACY	+/- 1 LSB
SETTLING TIME	100uS

## **DIGITAL PARALLEL USER PORT**

APPLICATIONS	user dependent
PROTOCOL	multiplexed byte serial 16 bit address 8 bit bidirectional data and control lines
LEVELS	HC TTL

## HIGH SPEED SYNCHRONOUS SERIAL PORT

APPLICATIONS	distributed processor
PROTOCOL	byte oriented, master
DATA RATE	56K baud
LEVELS	RS422

**Note 1:** Does not include probe nonconformability, however offset and gain error are removed during calibration.



## **XII PC1000 ERROR MESSAGES**

### **SRQ OVERFLOW**

An SRQ Overflow message is just to inform you that an interrupt has not been serviced. The interrupts are enabled and disabled in the SINT menu. If you enable an interrupt and don't service the interrupt, you get an SRQ Overflow. The message can be ignored without causing any problems or you can do one of the following: disable the interrupts in the SINT menu or service the interrupt in your program.