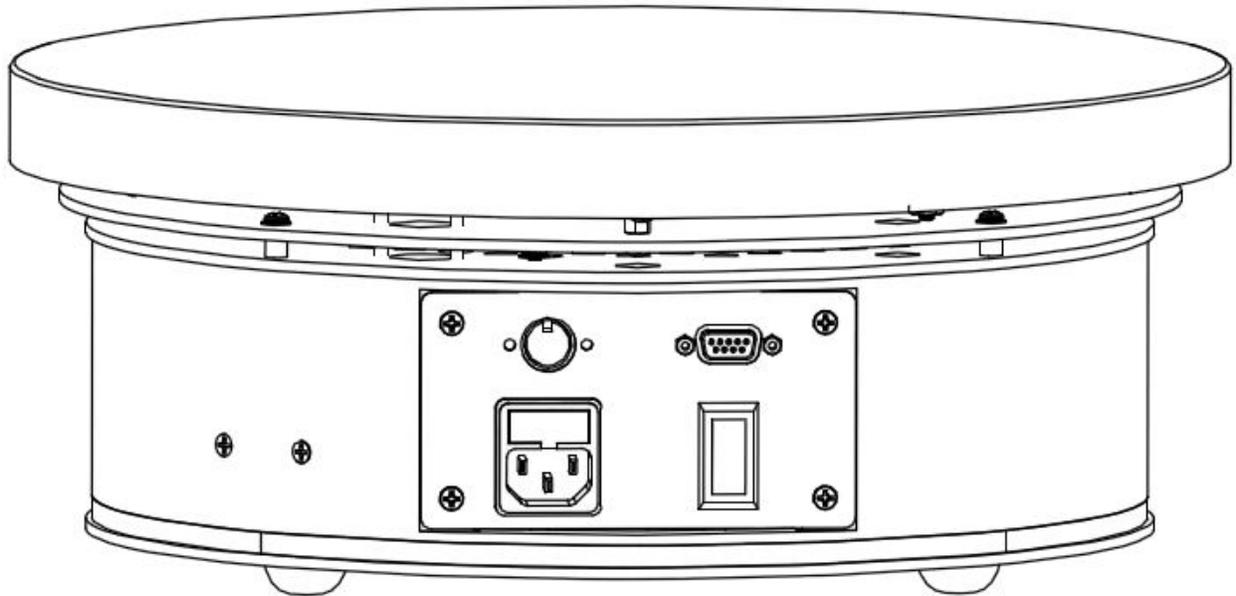


# HP90 Programming Manual

March 15, 2019

Applicable to HP90 firmware v1.0



**TORREY PINES  
SCIENTIFIC**

## Features

---

<b>Identification</b>	Unit Model and version Unit Serial Number User Defined String (10 chars max)
<b>Ramp To Set Point Temp</b>	Settable Ramp Rates: 0 – 450 C/hr (0=no ramping)
<b>Set Point (C)</b>	Settable Range: 10C to 350C “off” (Heater OFF mode—temp controller turned off)
<b>User Interface</b>	Default: RS232 Serial Port with Command Library LED Light Ring Temperature State Indicator: <ul style="list-style-type: none"><li>- Color Blue: Plate Temp below 30C or Unit Idle</li><li>- Color Yellow: Plate Temperature between 30C and 50C</li><li>- Color Red: Plate Temperature greater than 50C</li><li>- Solid Color: Plate is steady at Set Point Temp</li></ul> Optional External Hot Plate Controller provides LCD Display and Push Buttons
<b>Calibration</b>	Initially Calibrated at Factory 2 Point User Settable Measured Calibration Temperatures Entered to 0.1C Easily Resettable Back to Factory Values
<b>PID Control Values</b>	Standard Default Values User Settable Values
<b>Timer</b>	User Settable Start Time (format -- hh:mm:ss) Range: 00:00:00 to 99:59:59 Count Down ( 00:00:00 min) Count Up (99:59:59 max)
<b>Continuous Broadcast</b>	Control Temp (User settable period 0-99:59)
<b>Serial Event Notification</b>	TEMP_STEADY (User settable enable/disable) TIMER=0 (User settable enable/disable)
<b>Non-Volatile Memory</b>	User Settings are Stored and Recalled on Next Power-Up

## Manual Conventions

---

### **<CR>**

The notation “<CR>” refers to the ASCII character for “carriage return” which is decimal 13 or hex D. Every valid HP90 command must be terminated by this character. In HyperTerminal the character is sent when the “enter” key is pressed. Programs written in C for example, typically send this character when “\r” is appended to the transmitted command string.

### **<LF>**

The notation “<LF>” refers to the ASCII character for “line feed” or “new line” which is decimal 10 or hex A. Every string that is returned from the HP90 will be terminated with this character. Actually, every string that is returned from the HP90 will be terminated with a <CR> then a <LF>. In HyperTerminal the combined characters will cause the cursor to return to the beginning of the next line. Programs written in C for example, can use this character (often “\n”) for parsing returned strings.

### **(string)**

Characters within parenthesis are strings consisting of 7-bit ASCII characters and the string itself is the argument for a command. The length of the string is clarified in the discussion of the command and the two parentheses are not included in the transmission of the command string.

## Serial Interface

---

The HP90 line of products have an optional external controller with an LCD and push buttons that enable the user to set features and to monitor the several states during operation.

The unit is also capable of being controlled without the external controller using a simple RS232 serial interface. This document details the commands available and how to use them. Each command sent to the HP90 must be terminated with an ASCII carriage return character and every response string from the HP90 will be terminated with a character pair consisting of an ASCII carriage return (decimal 13) and ASCII linefeed character (decimal 10). All commands described within this manual are case sensitive. When a command is successfully received, an appropriate response string will be returned. If a command is received with a syntax error, the character “e” will be returned to indicate an error.

The command set was developed to provide functionality that will enable multiple interface and programming scenarios. For example, if a user wishes to use a simple application such as HyperTerminal to control a single unit, linefeed and carriage control characters included in the command and return strings enable clean formatting within the HyperTerminal User Interface. Additionally, commands are included to enable different programming methodologies such as polled response or event driven responses. For example, a program could be written to set a new set point then read the plate temperature at some time interval (polling) and take specific actions when the plate reaches various values along the way. Or a program could be written to set a new set point then the program could do something else until the set point is reached and the “TEMP\_STEADY” message is received from the unit (event driven). See “Command B” for setting/clearing event messages.

When the Serial COM settings are configured correctly on the external control computer, every properly formatted command sent to the HP90 will return either the requested data string or the string “ok” to acknowledge the successful reception of the command. The character “e” is returned when the HP90 does not recognize the received command string. The code running in the external computer can use the returned strings for handshaking and/or to verify that the command was properly executed.

If a serial terminal emulator application like HyperTerminal is used to send commands and receive responses, the command x<CR> will put the HP90 into “Terminal Mode”. Terminal Mode will enhance the screen data captured by HyperTerminal by displaying the HP90 response on the next line from the line that issued the command. Terminal Mode is cancelled when the HP90 is powered off.

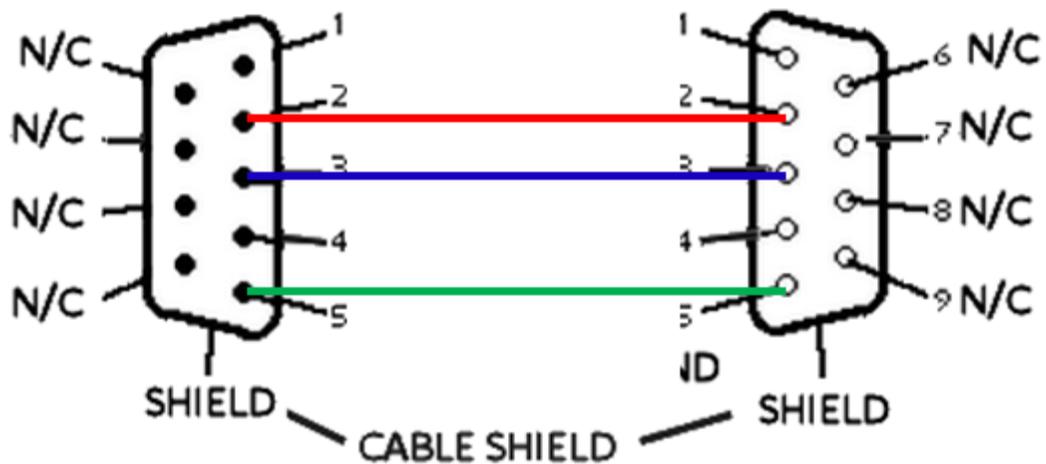
## COM Settings

---

9600 baud  
1 stop bit  
no parity  
no hardware handshake  
100ms delay after each command sent (after \r)

## Serial Cable Diagram

---



RS232 "Straight Through" Cable Pinout (Pins 2,3,5 Only)

## Unit Identification Commands

---

### Command: v

#### Function: **Return HP90 Model and Version**

Description: When the command v<CR> is received by the HP90 unit, the model number and the firmware version will be returned in a text string terminated by <CR><LF>. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example:

send: v<CR>

returned: HP90 v1.00 <CR><LF>

### Command: V

#### Function: **Return Serial Number**

Description: When the command V<CR> is received by the HP90 unit, the 8 character serial number will be returned in a text string terminated by <CR><LF>. The serial number for every HP90 unit is unique. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Return the 8 char serial number:

send: V<CR>

returned:12345678 <CR><LF>

### Command: >(user defined string)

#### Function: **Store User Defined String**

Description: Each unit has a unique serial number (ref cmd V) but it may be helpful to assign a custom “name” to identify a unit. When the command “Return User Defined String” (cmd “>”) is issued, the User Defined String will also appear on the screen of the LCD which provides the ability to notify the user when action is required. The screen will return to previous screen when the user presses either the up or down arrow buttons on the unit or when the command char “^” is issued over the serial line. This command enables a string of up to 10 characters to be stored with the unit. When the command >(user defined string)<CR> is received by the HP90 unit, the string is stored and the unit returns ok<CR><LF>. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example—assign HP90 the name “Unit 1”

send: Unit 1<CR>

returned: ok <CR><LF>

Example—verify that the name is “Unit 1”

send: ><CR>

returned: Unit 1<CR><LF>

**Command: >**

**Function: Return User Defined String**

Description: When the command ><CR> is received by the HP90 unit, the User Defined String will be returned in a text string terminated by <CR><LF>. If no string has been stored, 10 space characters will be returned followed by <CR><LF>. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example—Return the string that was stored using the >(user defined string) command:

send: ><CR>

returned: Unit 1<CR><LF> if “Unit 1” was previously stored  
“Unit 1” is also displayed on LCD screen of HP90

returned: <CR><LF> if no user string has been stored

LCD screen will be blank if no user string has been stored (or user string is the space char)

## Temperature Commands

---

**Command: s**

**Function: Return Set Point Temperature**

Description: When the command s<CR> is received by the HP90 unit, the current set point temperature will be returned in a text string terminated by <CR><LF>. The setpoint format length is variable depending on the number of digits in the setpoint temperature and whether the value is negative. The value will always include a decimal point and one character representing the fractional tenth value. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Note: Putting the HP90 into Heater OFF Mode (see “Command: i”) will turn off the temperature controller and the set point will be read as “off”.

Example—Return the current set point:

send: s<CR>

returned example 1: 20<CR><LF>

returned example 2: 250<CR><LF>

returned example 3: off<CR><LF> if unit is in Heater OFF Mode (see “Command: i”)

### Command: n

#### Function: **Set and Store New Set Point Temperature**

Description: When the command n(new\_temperature)<CR> is received by the HP90 unit, the set point will be changed to new\_temperature and the text string “ok” will be returned terminated by <CR><LF>. The format for new\_temperature is variable depending on the number of digits in the desired new set point temperature and whether the value is negative. The value must always include a decimal point and one character representing the fractional tenth value. The settable range for an HP90 is 10C to 350C. A value of 0 (zero) may also be sent to set the “Heater OFF” mode.

**NOTE: Use of the ramp feature always requires entering a ramp value prior to entering a set point temperature. If a ramp rate is required that is different from the currently stored ramp rate, the ramp rate must be changed before a new set point temperature is entered. If a different ramp rate is set after a new set point temperature has been entered, the HP90 will heat or cool per the previous ramp rate—not per the different ramp rate. See “Command: L(ramp\_rate)”**

If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example1— Change set point to 50C:

send: n50<CR>

returned: ok <CR><LF>

### Command: i

#### Function: **Set HP90 Unit to Heater OFF Mode**

Description: When the command i<CR> is received by the HP90 unit, the temperature controller will be switched “off”, meaning that the plate will no longer heat or cool. In Heater OFF Mode, the unit will report a set point of “off”. To exit Heater OFF Mode (ie: turn the heater back on), use command I (see below) or set the set point to a new value using cmd n. If the command is not received in the proper syntax, e<CR><LF> will be returned.

The table below provides an example to illustrate the results depending on the various methods of setting an clearing Heater OFF mode.

Set Point Was	Turn Heater OFF	Turn Heater ON	Set Point Now
125	i<CR>	I<CR>	125
		n100<CR>	100
	n0<CR>	I<CR>	20
		n100<CR>	100

Example – Set Heater OFF Mode:

send: i<CR>

returned: ok <CR><LF>

Example – verify Heater OFF Mode:

send: s<CR>

returned: off<CR><LF>

Example – exit Heater OFF Mode by setting new set point to 50C

send: n50<CR>

returned: ok <CR><LF>

Example – verify no longer in Heater OFF Mode and set point is 50C:

send: s<CR>

returned: 50<CR><LF>

### Command: I

Function: **Clear HP90 Heater OFF Mode and Return to Last Setpoint**

Description: The command I<CR><LF> will exit Heater OFF Mode and return the HP90 setpoint to the value that it was before the unit entered Heater OFF Mode. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – verify Heater OFF Mode:

send: s<CR>

returned: off<CR><LF>

Example – Clear Heater OFF Mode (setpoint was 50C before entering Heater OFF Mode):

send: I<CR>

returned: ok <CR><LF> (setpoint now 50C)

Example – verify no longer in Heater OFF Mode and set point is 50C:

send: s<CR>

returned: 50<CR><LF>

### Command: L

Function: **Return the current Ramp Rate**

Description: When the command L<CR> is received by the HP90 unit, the ramp rate will be returned in a text string terminated by <CR><LF>. The returned value will have 1 to 3 chars representing 0 to 450C/hr (“0” represents “no ramp” or “ramping disabled”). If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Determine the current ramp rate:

send: L<CR>

returned: 100 <CR><LF> ( the current ramp rate is 100C/hr)

**Command: L(ramp\_rate)**

**Function: Set and Store the Ramp Rate**

Description: When the command L(ramp\_rate)<CR> is received by the HP90 unit, the ramp rate will be set and the text string "ok" will be returned terminated by <CR><LF>. The format for ramp\_rate is 1 to 3 chars to represent values of 0 to 450C/hr. A value of "0" represents "no ramp" or "ramping disabled".

If the command is not received in the proper syntax, e<CR><LF> will be returned.

**NOTE: Use of the ramp feature always requires entering a ramp value prior to entering a set point temperature. If a ramp rate is required that is different from the currently stored ramp rate, the ramp rate must be changed before a new set point temperature is entered. If a different ramp rate is set after a new set point temperature has been entered, the HP90 will heat or cool per the previous ramp rate—not per the different ramp rate.**

Example1— Change/Set the ramp rate to 100C/hr:

send: L100<CR>

returned: ok <CR><LF>

Example2— Turn off ramping (heat or cool at maximum rate)

send: L0<CR>

returned: ok <CR><LF>

**Command: p**

**Function: Return the current Plate Temperature**

Description: When the command p<CR> is received by the HP90 unit, the current plate temperature will be returned in a text string terminated by <CR><LF>. The format length is variable depending on the number of digits in the plate temperature and whether the value is negative. The value will always include a decimal point and one character representing the fractional tenth value. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example—Return the plate temperature:

send: p<CR>

returned: 50 <CR><LF>

If an issue occurs with the temperature measurement circuitry or with a calibration value, the unit will return the error code in place of the plate temperature value. If an error is detected, the unit will go into Heater OFF Mode (power to the heater plate will be turned off). The table below lists the error codes, the likely cause, and recommended action to address the error. Calibration errors (cal0-cal4) are easily addressed by performing the action to reset the appropriate calibration points to the factory settings. The unit can then be re-calibrated if necessary.

<b>Error Code</b>	<b>Possible Cause</b>	<b>Recommended Action</b>
RTD <sub>0</sub>	The RTD Sensor is not connected or has failed	Inspect Sensor and/or Contact Torrey Pines Scientific
RTD <sub>s</sub>	The RTD Sensor has shorted or has failed	
cal0	The Calibrated Temperature Value is out of range	Reset High, Low, or Both Cal Points
cal1	Low Cal Point out of range	Reset Low Cal Point
cal2	High Cal Point out of range	Reset High Cal Point
cal3	High Point Measured Cal Value is Lower than Low Point Measured Value (or reverse)	Reset High, Low, or Both Cal Points
cal4	High Point Temperature Value is Lower than Low Point Temperature Value (or reverse)	

## Timer Commands

---

### Command: **a**

#### Function: **Return Current Timer Value**

Description: When the command a<CR> is received by the HP90 unit, the current timer value will be returned in a text string terminated by <CR><LF>. The timer string is a fixed width of 8 characters in the format hh:mm:ss. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – the current timer value is 1 hour, 32 minutes, 15 seconds:

send: a<CR>

returned: 01:32:15 <CR><LF>

Example – verify that the current timer has been cleared (see Command: ac):

send: a<CR>

returned: 00:00:00 <CR><LF>

### Command: **a(hh:mm:ss)**

#### Function: **Set Current Timer Value**

Description: When the command a(hh:mm:ss)<CR> is received by the HP90 unit, the current timer value will be set to the new value in the 8 character format hh:mm:ss. The settable timer range is 00:00:00 to 99:59:59. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Set the current timer value to 1 hour, 32 minutes, 15 seconds:

send: a01:32:15<CR>

returned: ok <CR><LF>

**Command: au**

**Function: Start Timer—Count Up**

Description: When the command au<CR> is received by the HP90 unit, the current timer value will increment every second. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Verify current timer value is 00:00:00

send: a<CR>

returned: 00:00:00 <CR><LF>

Example – Start incrementing timer

send: au<CR>

returned: ok<CR><LF>

Example – Check timer after 5 seconds

send: a<CR>

returned: 00:00:05 <CR><LF>

**Command: ad**

**Function: Start Timer—Count Down**

Description: When the command ad<CR> is received by the HP90 unit, the current timer value will decrement every second. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Verify current timer value is 00:30:00

send: a<CR>

returned: 00:30:00 <CR><LF>

Example – Start decrementing timer

send: ad<CR>

returned: ok<CR><LF>

Example – Check timer after 5 seconds

send: a<CR>

returned: 00:29:55 <CR><LF>

**Command: ap**

**Function: Pause or Stop Timer**

Description: When the command ap<CR> is received by the HP90 unit, the current timer value will stop decrementing or incrementing. To restart the timer from the current timer value, send either an au or ad command. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Stop Timer

send: ad<CR>

returned: ok<CR><LF>

Example – Read Current Timer Value

send: a<CR>

returned: 00:29:55 <CR><LF>

**Command: ac**

**Function: Clear the Current Timer Value**

Description: When the command ac<CR> is received by the HP90 unit, the current timer value will be set to 00:00:00. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Clear the Current Timer Value

send: ac<CR>

returned: ok<CR><LF>

Example – Read Current Timer Value

send: a<CR>

returned: 00:00:00 <CR><LF>

## Calibration Commands

---

**Command for Plate: R**

**Function: Return the High Calibration Point Temperature**

Description: When the command R<CR> is received by the HP90 unit, the temperature at which the high calibration point was established will be returned in a text string terminated by <CR><LF>. The high calibration point temperature is the set point temperature when the high temperature calibration was performed. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Determine the high temperature calibration point for the Plate:

send: R<CR>

returned: 250<CR><LF> ( the HP90 was calibrated at 250C)

**Command for Plate: r**

**Function: Return the Low Calibration Point Temperature**

Description: When the command r<CR> is received by the HP90 unit, the temperature at which the low calibration point was established will be returned in a text string terminated by <CR><LF>. The Low calibration point temperature is the set point temperature when the low temperature calibration was performed. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Determine the low temperature calibration point for the plate:

send: r<CR>

returned: 50 <CR><LF> ( the HP90 was calibrated at 50C)

### Command for Plate: T

Function: **Return the Measured Temperature at the High Calibration Point**

Description: When the command T<CR> is received by the HP90 unit, the measured temperature when the HP90 was at the high calibration point temperature will be returned in a text string terminated by <CR><LF>. The measured temperature format length is variable depending on the number of digits in the calibration temperature and whether the value is negative. The value will always include a decimal point and one character representing the fractional tenth value. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Determine the measured temperature at the high calibration point of 250C:  
send: T<CR>  
returned: 251.7 <CR><LF> ( the plate was measured to be 251.7C when the HP90 was set to 250C)

### Command for Plate: T(measured\_temp)

Function: **Set the Measured Temperature at the High Calibration Point**

Description: When the command T(measured\_temp)<CR> is received by the HP90 unit, measured\_temp will be stored and used with the High Calibration Point temperature to calculate calibration offsets. The measured\_temp format length is variable depending on the number of digits in the measured temperature. The value will always include a decimal point and one character representing the fractional tenth value. If the command is not received in the proper syntax, e<CR><LF> will be returned.

**\*\*IMPORTANT NOTE:** Both the HP90 temperature and the calibration measurement temperature must be steady at the High Calibration Point for at least 10 minutes before the measured temperature is entered. Additional settling time may be required depending on the thermal conduction of the material at the calibration measurement point. For example, if a liquid sample is the measurement point for the calibration, more time may be required to reach a stable measurement temperature. If the unit or the measured value are not steady at the High Calibration Point temperature when the measured value is entered, significant calibration error may result.

Example – The HP90 High Calibration Point is 250C, the measured temp when the unit is steady at 250C is 251.7 C. Set the measured High Calibration Point temperature for the plate:

send: T251.7 <CR>  
returned: ok <CR><LF>

Example – Verify that the Measured Temperature at the High Cal Point is 251.7C  
send: T<CR>  
returned: 251.7 <CR><LF>

#### Command for Plate: **t**

##### Function: **Return the Measured Temperature at the Low Calibration Point**

Description: When the command t<CR> is received by the HP90 unit, the measured temperature when the HP90 was at the low calibration point temperature will be returned in a text string terminated by <CR><LF>. The measured temperature format length is variable depending on the number of digits in the calibration temperature and whether the value is negative. The value will always include a decimal point and one character representing the fractional tenth value. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – Determine the measured temperature of the plate at the low calibration point of 50C:

send: t<CR>

returned: 48.6 <CR><LF> ( the plate was measured to be 48.6C when the HP90 was set to 50C)

#### Command for Plate: **t(measured\_temp)**

##### Function: **Set the Measured Temperature at the Low Calibration Point**

Description: When the command t(measured\_temp)<CR> is received by the HP90 unit, measured\_temp will be stored and used with the Low Calibration Point temperature to calculate calibration offsets. The measured\_temp format length is variable depending on the number of digits in the measured temperature. The value will always include a decimal point and one character representing the fractional tenth value. If the command is not received in the proper syntax, e<CR><LF> will be returned.

**\*\*IMPORTANT NOTE:** Both the HP90 temperature and the calibration measurement temperature must be steady at the Low Calibration Point for at least 10 minutes before the measured temperature is entered. Additional settling time may be required depending on the thermal conduction of the material at the calibration measurement point. For example, if a liquid sample is the measurement point for the calibration, more time may be required to reach a stable measurement temperature. If the unit or the measured value are not steady at the Low Calibration Point temperature when the measured value is entered, significant calibration error may result.

Example – The HP90 Low Calibration Point for the plate is 50C, the measured temp of the plate surface when the unit is steady at 50C is 48.6C. Set the measured Low Calibration Point temperature for the plate:

send: t48.6<CR>

returned: ok <CR><LF>

Example – Verify that the Measured Temperature at the Low Cal Point is 48.6

send: t<CR>

returned: 48.6<CR><LF>

**Command for Plate: H**

Function: **Reset the High Temperature Calibration Points to Factory Value**

Description: When the command H<CR> is received by the HP90 unit, the High Temperature Calibration Point and the Measured Temperature at the High Calibration Point will be set to the high point value set during factory calibration. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – set the High Calibration Point for the plate to Factory Value

send: H<CR>

returned: ok <CR><LF>

**Command for Plate: h**

Function: **Reset the Low Temperature Calibration Points to Factory Value**

Description: When the command h<CR> is received by the HP90 unit, the Low Temperature Calibration Point and the Measured Temperature at the Low Calibration Point will be set to the low point value set during factory calibration. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – set the Low Calibration Point for the plate to Factory Value

send: h<CR>

returned: ok <CR><LF>

## Macro Commands

---

### Command for Plate: **m**

#### Function: **Macro to return all 4 calibration values**

Description: When the command m<CR> or k<CR> is received by the HP90 unit, the Low Temperature Calibration Point, the Measured Temperature at the Low Calibration Point, the High Temperature Calibration Point, and the Measured Temperature at the High Calibration Point will be returned in that order in a string delimited by commas and terminated by <CR><LF>. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – return all plate cal points from the previous examples

send: m<CR>

returned: 50,48.6,250,251.7 <CR><LF>

### Command: **M**

#### Function: **Macro to return all Status, SP, PT, Timer**

Description: When the command M<CR> is received by the HP90 unit, a string containing the Status String, Set Point Temperature, the Temperature, and the Current Timer Value will be returned delimited by commas and terminated by <CR><LF>. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – return Status, SP, PT, Timer

send: M<CR>

returned: StbLH,50,50,00:04:13<CR><LF> (temp is steady, timer not running, unit not broadcasting, low temp cal done, high temp cal done, set point is 50, plate temp is 50, the timer value is 4 mins 13 sec)

## Event Notification Commands

---

### Command: **b(mm:ss)**

#### Function: **Broadcast the Control Temperature**

Description: When the command b(mm:ss)<CR> is received by the HP90 unit, the unit will return the currently specified control (plate or probe) temperature every time interval defined by mm:ss. To disable the Plate Temperature, set the time interval to 00:00. Sending b<CR> will return the current settings. If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – return the plate temperature every 5 seconds

send: b00:05<CR>

returned: ok <CR><LF>

Example – stop broadcasting the plate temperature

send: b00:00<CR>

returned: ok <CR><LF>

Example – return the current Broadcast Plate Temperature settings

send: b<CR>

returned: 00:00 <CR><LF>

### Command: **B(sz)**

Function: **Broadcast Temperature or Timer Events**

Description: When the command B(ab)<CR> is received by the HP90 unit, the unit will enable or disable Temperature or Timer Events depending on the case of the variables s and z as shown in the table below:

Return string “TEMP\_STEADY<CR><LF>” when plate temperature is steady\*

S = enable

s = disable

Return string “TIMER=0<CR><LF>” when timer reaches zero

Z = enable

z = disable

Sending B<CR> will return the current settings.

If the command is not received in the proper syntax, e<CR><LF> will be returned.

Example – set to broadcast when the temperature is steady at the set point but do not broadcast when the timer has reached zero.

send: BSz<CR>

returned: ok <CR><LF>

Example – return the current Broadcast Event settings

send: B<CR>

returned: Sz <CR><LF>

\* The “Temperature is Steady” condition is set when the control temperature has been within +/- 0.2C for 60 seconds. This condition has been shown to be sufficient for typical user applications. Large thermal loads, changing thermal loads, or other environmental conditions can cause the control temperature to vary outside of the +/- 0.2C/60sec boundaries resulting in multiple “Temperature is Steady” messages from the HP90. In these situations, the user should characterize their system then determine

their own “temperature is steady” criteria by continuously reading the control temperature over an appropriate time period instead of using the “TEMP\_STEADY” flag.

**Command:** Y, y

**Function:** Turn the beeper on the HP90 On or Off

**Description:** The HP90 unit has a piezoelectric beeper that is used as an audible signal for front panel key presses and for timer alarm conditions if enabled (Menu/Timer Options). Beeper can be turned on and off remotely using the command Y<CR> to turn on and y<CR> to turn off.

**Example – turn the beeper on then off**

send: Y<CR>

returned: ok<CR><LF> (beeper is now on—nearby people are annoyed)

send: y<CR>

returned: ok<CR><LF> (beeper is now off—nearby people are happy again)

## Utility Commands

---

**Command:** x

**Function:** Set “terminal mode”

**Description:** When the command x<CR> is received by the IC50 unit, the unit will return a string consisting of x <CR><LF> ok <CR><LF> that indicate that the unit has enabled “terminal mode”. When the unit is in terminal mode, the pair <CR><LF> will be returned from the IC50 immediately after every <CR> char is received by the IC50. This will enable a serial terminal application like or similar to HyperTerminal to advance the position of the cursor by one line so that the next string of characters returned by the IC50 will not over-write the current line. This is useful to log every command sent and every response received.

Terminal mode will be cancelled when the IC50 unit is powered “off” then back “on”.

**Example – set “terminal mode”**

send: x<CR>

returned: x <CR><LF> ok <CR><LF> (unit is now in terminal mode)

**Default power on mode serial application display example:**

send commands n25.0 <CR>, s <CR>

In the default power up mode, the serial screen application (HyperTerminal) will display as shown below:

ok50 ← n250<CR> sent, “ok”<CR><LF> is returned and over-writes “n250”

250 ← s<CR> sent, “250<CR><LF> is returned and over-writes “s”

After sending the “Set terminal mode” command (x), the serial screen application (HyperTerminal) will display as shown below:

n250 ← n250<CR> sent

ok ← “ok” returned on next line

s      ← s<CR> sent  
250    ← "25.0" returned on next line

### Command: **S**

#### Function: **Return Status String**

Description: When the command S<CR> is received by the HP90 unit, the unit will return a string of 5 characters in the order stblh<CR><LF> that indicate the current status of the unit as described in the table below:

s = temperature is not steady  
S = temperature is steady\*  
t = timer is not running  
T = timer is running  
l = Plate low temp cal has not been changed by user (Factory values stored)  
L = Plate low temp cal has been changed by user  
h = Plate high temp cal has not been do changed by user ne (Factory values stored)  
H = Plate high temp cal has been changed by user

Example – return the Status String

send: S<CR>

returned: StbLH <CR><LF> (temp is steady, timer not running, unit not broadcasting, low temp cal done, high temp cal done)

\* "Temperature is Steady" condition is set when the control temperature has been within +/- 0.2C for 60 seconds. This condition has been shown to be sufficient for typical user applications. Large thermal loads, changing thermal loads, or other environmental conditions can cause the control temperature to vary outside of the +/-0.2C/60sec boundaries resulting in multiple "Temperature is Steady" messages from the HP90. In these situations, the user should characterize their system then determine their own "temperature is steady" criteria by continuously reading the control temperature over an appropriate time period instead of using the "TEMP\_STEADY" flag.

**Command: #Z**

**Function: Reset Unit to Default Configuration**

Description: When the command #Z<CR> is received by the HP90 unit, the unit will reset all non-volatile memory to default values. All calibrations, user settings, and PID values will revert to factory settings or default. In typical operation, the “#Z” function will rarely if ever be necessary. The default values are shown in the table below:

Parameter	Default Value
Set Point	20
Ramp Rate	360
Cal Point Low	factory
Cal Point High	factory
Timer	0:00:00
Broadcast Period	0:00
Broadcast Events	none
Status Byte	0
User String	cleared

## Serial Command Quick Reference Table

			Example	
	Command	Function	sent	returned
ID	v	return model and version	v <CR>	HP90 v1.0 <CR LF>
	V	return serial number	V <CR>	12345678 <CR LF>
	>(abc123...)	set user string (10 chars max)	>UNIT 10 <CR>	ok <CR LF>
	>	return user string	> <CR>	UNIT 10 <CR LF>
Temperature	s	return set point temperature	s <CR>	150 <CR LF> off <CR LF> if Heater OFF Mode
	n(xxx)	set new set point temperature (10min-350max)	n150<CR>	ok <CR LF>
	i/l	set/clear Heater OFF Mode (plate power is off when set)	i <CR> or l<CR>	ok <CR LF>
	L	return ramp rate (C/hr)	L<CR>	50 <CR LF>
	L(xxx)	set ramp rate (C/hr) 0=no ramp, 450 max	L50<CR>	ok <CR LF>
	p	return plate temperature	p <CR>	50 <CR LF>
Timer	a	return current timer value	a <CR>	00:04:13 <CR LF>
	a(hh:mm:ss)	set timer value (24:59:59 max)	a00:05:00 <CR>	ok <CR LF>
	au	start timer--count up	au <CR>	ok <CR LF>
	ad	start timer--count down	ad <CR>	ok <CR LF>
	ap	pause/stop timer	ap <CR>	ok <CR LF>
	ac	clear timer (00:00:00)	ac <CR>	ok <CR LF>
	M	macro to return status,sp,plate temp, timer	M <CR>	StbLH,150,150,00:04:13 <CR LF>
Calibration	R	return HIGH cal point temperature	R<CR>	175 <CR LF>
	r	return LOW cal point temperature	r<CR>	50 <CR LF>
	Txxx.x)	set measured temperature at HIGH cal point (x.x min)	T149.3<CR>	ok <CR LF>
	T	return measured RTD temp at HIGH cal point	T<CR>	149.3<CR LF>
	txxx.x)	set measured temperature at LOW cal point (x.x min)	t51.7<CR>	ok <CR LF>
	t	return measured RTD temp at LOW cal point	t<CR>	51.7<CR LF>
	H	reset HIGH temp cal points to Factory high cal value	H<CR>	ok <CR LF>
	h	reset LOW temp cal points to Factory low cal value	h<CR>	ok <CR LF>
Event Notification	m	macro to return all 4 cal values (r,t,R,T)	m <CR>	50.2,51.7,150.7,149.3 <CR LF>
	b(mm:ss)	Broadcast Control Temperature returns Control Temp every mm:ss, 99:59 max	b00:10 <CR>	plate temp <CR LF> returned every 10 seconds
	B(sz)	Broadcast Temperature or Timer Events Return "TEMP_STEADY" when temp is steady S = enable    s = disable Return "TIMER=0" when timer reaches zero Z = enable    z = disable	BSz <CR>	TEMP_STEADY <CR LF> returned at event  TIMER=0 <CR LF> not returned at event
Utility	Y/y	Turn on/off audible beeper in unit	Y<CR> or y<CR>	ok<CR><LF> {Y: Unit beeper On}
	x	Set Terminal Mode to format output for HyperTerminal	x<CR>	ok <CR><LF>
	S	return status S = temperature is (s = is not) steady T = timer is (t = is not) running L = plate low cal has (l = has not*) been changed H = plate high cal has (h = has not*) been changed	S <CR>	StLH <CR LF> --temp is steady --timer not running --plate cal changed at low temp --plate cal changed at high temp
	#Z	Reset unit to default configuration (rarely necessary)	#Z<CR>	Unit will reset

<CR> is return char (for example: "enter" keyboard press for HyperTerminal, "\r" for C pgms, ASCII hex char "D")

\*Factory Calibration Values are Stored

## Appendix A: User Settable PID Values

The temperature of the HP90 unit is controlled using a PID algorithm responding to input from the platinum RTD temperature sensor in the plate or the platinum RTD temperature sensor in the probe (depending on the control method selected for the unit—see “Command: c”). The default PID constants for both the plate and the probe have been developed to perform across a wide range of anticipated user applications.

However, if a user determines that their specific application may benefit from adjusting one or more of the PID constants, the ability to change these constants is provided in the HP90 unit via the serial line. The table below lists the commands to get and set the values for the three constants for both the plate and the probe control loops. Additionally, a command is provided to reset the constants to the default values.

### WARNING:

**Changing the PID values should only be performed by those who are thoroughly knowledgeable in the operation and behavior of PID control loops and the characteristics of the physical systems that they are attempting to control. If not done correctly, changing a value or combination of values may have a dramatically detrimental effect on the ability of the HP90 to control to the set point temperature.**

		Example		
Command	Function	sent	returned	
Plate	#kp	Return current value for proportional constant	#kp <CR>	300<CR LF>
	#ki	Return current value for integral constant	#ki <CR>	100 <CR LF>
	#kd	Return current value for differential constant	#kd <CR>	450 <CR LF>
	#kp(int)	Set value for proportional constant. Value must be non-negative integer less than 9999999. (example: 15000)	#kp300 <CR>	ok <CR LF>
	#ki(int)	Set value for proportional constant. Value must be non-negative integer less than 9999999. (example: 250)	#ki100 <CR>	ok <CR LF>
	#kd(int)	Set value for proportional constant. Value must be non-negative integer less than 9999999. (example: 50000)	#kd450 <CR>	ok <CR LF>
	#k0	Reset kp,ki, and kd to default values	#k0 <CR> (“0” is zero)	ok <CR LF>

<CR> is return char (for example: “enter” keyboard press for HyperTerminal, “\r” for C pgms, ASCII hex char “D”)

[ ] Commands in brackets are for probe (do not include brackets in sent command chars)

\*Default Values Stored