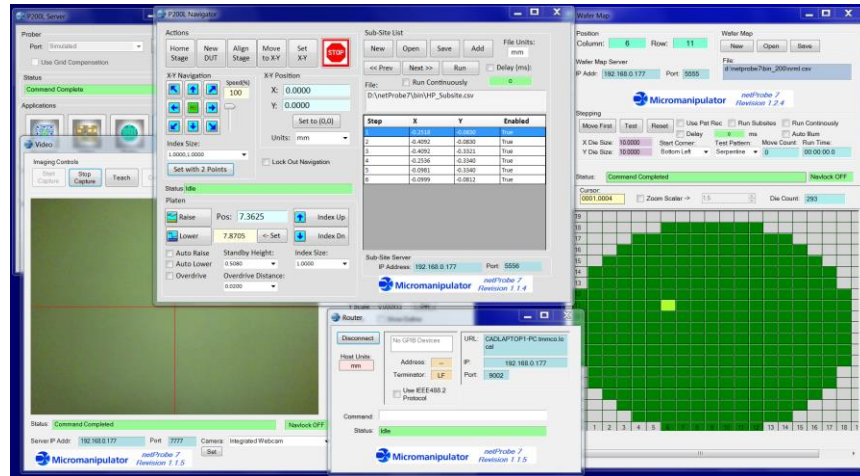


# netProbe 7 Software Suite for P200L Probing Systems

P/N: A1018225 Rev. C 3 December 2020



© Copyright 2020 by



# Micromanipulator

1555 Forrest Way

Carson City, Nevada, 89706

[sales@micromanipulator.com](mailto:sales@micromanipulator.com)

PRINTED IN USA

## Table of Contents

1	netProbe7 Software Suite for P200L.....	8
1.1	Software components.....	8
1.1.1	Applications.....	9
1.1.2	Libraries.....	9
1.2	Computer Login.....	9
1.3	GPIB Connections.....	10
1.3.1	Test Host connected to P200L <b>without netProbe 7 and netProbe 7 PC</b> .....	10
1.3.2	Tester connected to P200L without netProbe 7 and netProbe 7 PC.....	11
1.3.3	Tester/Test Host <b>connected to netProbe 7/netProbe 7 PC</b> .....	12
1.3.4	Tester connected to P200L <b>with netProbe 7 and netProbe 7 PC</b> .....	13
2	P200L_Server .....	14
2.1	Startup .....	16
2.2	Shut Down.....	18
2.3	TCP/IP Configuration.....	18
3	netProbe7 P200L Navigator .....	19
3.1	Getting Started.....	20
3.1.1	Startup.....	20
3.1.2	Recommended Setup Sequence .....	21
3.1.3	Home Positioners .....	21
3.1.4	Loading a Wafer or DUT.....	22
3.1.5	Aligning X-Y Stage.....	23
3.1.6	Z Setup .....	24
3.2	Coordinate Positioning.....	26
3.2.1	X-Y Indexing with Navigation Jog Buttons .....	26
3.2.2	Directed X-Y Moves.....	27
3.3	Navigator Reference .....	28
3.3.1	X-Y Stage .....	28
3.3.2	Set X-Y Index Using Two Points.....	29
3.3.3	Z Platen .....	30
3.4	Actions .....	31
3.4.1	Home.....	31

3.4.2	New DUT .....	31
3.4.3	Align Stage.....	31
3.4.4	Move to X-Y.....	31
3.4.5	Set X-Y .....	32
3.4.6	Sub-Site List.....	33
4	netProbe7 Wafer Map .....	36
4.1	Overview .....	37
4.1.1	Wafer Map Window.....	37
4.2	Create a new map .....	39
4.2.1	Set Die Size .....	39
4.2.2	Selecting a map type.....	40
4.2.3	Generating the Map.....	40
4.3	Wafer Map File Format .....	40
5	netProbe7 Video .....	44
5.1	Startup .....	44
5.1.1	Camera Selection .....	44
5.1.2	Live Video .....	44
5.1.3	Setting up Pattern Recognition.....	45
5.1.4	Functions using Mouse/Key Combinations.....	47
5.2	Reference .....	48
5.2.1	Imaging Controls .....	48
5.2.2	Templates.....	51
6	netProbe 7 Router.....	53
7	Application Command Sets .....	57
7.1	P200L Server TCP/IP Commands.....	57
7.1.1	Host Command Response Codes .....	57
7.1.2	Asynchronous Update Format .....	58
7.1.3	netProbe 7 P200L Server TCP/IP Command Set .....	60
7.1.4	*IDN?.....	62
7.1.5	:JOY:IL.....	62
7.1.6	:JOY:IL? .....	62
7.1.7	:JOY:NAVLOCK.....	63

7.1.8	:JOY:NAVLOCK?	63
7.1.9	:PRB:AL	64
7.1.10	:PRB:AL? Get mechanical alignment angle in radians.	64
7.1.11	:PRB:CEN:ALL	65
7.1.12	:PRB:CEN:VAL	65
7.1.13	:PRB:CEN:VAL?	66
7.1.14	:PRB:CEN:X	66
7.1.15	:PRB:CEN:Y	67
7.1.16	:PRB:DN	67
7.1.17	:PRB:DNM	68
7.1.18	:PRB:DNM? Get the auto-z modes for the platen.	68
7.1.19	:PRB:GRID:FILE	69
7.1.20	:PRB:GRID:FILE? Get the grid file name	70
7.1.21	:PRB:GRID:USE	70
7.1.22	:PRB:GRID:USE?	71
7.1.23	:PRB:HO:ALL	71
7.1.24	:PRB:HO:ALL?	72
7.1.25	:PRB:HO:X Home the X axis.	72
7.1.26	:PRB:HO:X?	72
7.1.27	:PRB:HO:XY	73
7.1.28	:PRB:HO:XY?	73
7.1.29	:PRB:HO:Y	74
7.1.30	:PRB:HO:Y?	74
7.1.31	:PRB:HO:Z	74
7.1.32	:PRB:HO:Z?	75
7.1.33	:PRB:JOG:XY	75
7.1.34	:PRB:JOG:Z	76
7.1.35	:PRB:KI	76
7.1.36	:PRB:LIM? Get the state of the XYZ limits	77
7.1.37	:PRB:MOV:XYA	77
7.1.38	:PRB:MOV:XYNO	78
7.1.39	:PRB:MOV:XYR	78

7.1.40	:PRB:MOV:XYRE.....	78
7.1.41	:PRB:MOV:ZA .....	79
7.1.42	:PRB:MOV:ZRE.....	79
7.1.43	:PRB:OR .....	80
7.1.44	:PRB:OR? .....	80
7.1.45	:PRB:POS:XYZ?.....	81
7.1.46	:PRB:POS:XYZNO? .....	81
7.1.47	:PRB:POS:XYZR?.....	82
7.1.48	:PRB:REFXY .....	82
7.1.49	:PRB:REFXY? .....	83
7.1.50	:PRB:STO.....	83
7.1.51	:PRB:TRA?.....	83
7.1.52	:PRB:UDO .....	84
7.1.53	:PRB:UDO?.....	84
7.1.54	:PRB:UN.....	85
7.1.55	:PRB:UN:VAL.....	85
7.1.56	:PRB:UN:VAL?.....	86
7.1.57	:PRB:UP .....	86
7.1.58	:SYS:OWN .....	87
7.1.59	:SYS:OWN? Get the owned state and the owner .....	87
7.1.60	:SYS:REG? .....	88
7.1.61	ERR? .....	88
7.1.62	STA? Get current motion control status .....	89
7.1.63	UPDT.....	89
7.1.64	UPDT?.....	90
7.2	Navigator TCP/IP Commands.....	91
7.2.1	:SUB:CURR? .....	91
7.2.2	:SUB:FILE Open a sub-site file .....	91
7.2.3	:SUB:FILE?.....	92
7.2.4	:SUB:FIRST? .....	92
7.2.5	:SUB:GOTO .....	93
7.2.6	:SUB:NEXT? .....	93

7.2.7	:SUB:PREV?.....	94
7.2.8	:SUB:RESET .....	95
7.2.9	:SYS:OWN .....	95
7.2.10	:SYS:OWN? .....	95
7.3	Wafer Map TCP/IP Commands .....	97
7.3.1	:SYS:OWN .....	98
7.3.2	:SYS:OWN? .....	98
7.3.3	:WFR:AUTOIL.....	99
7.3.4	:WFR:AUTOIL?.....	99
7.3.5	:WFR:BIN .....	99
7.3.6	:WFR:BIN:LIST?.....	100
7.3.7	:WFR:BIN:VAL.....	100
7.3.8	:WFR:CURR? .....	101
7.3.9	:WFR:DIESIZE.....	101
7.3.10	:WFR:DIESIZE? .....	102
7.3.11	:WFR:FILE .....	102
7.3.12	:WFR:FILE?.....	103
7.3.13	:WFR:FIRST:VAL?.....	103
7.3.14	:WFR:FIRST? .....	104
7.3.15	:WFR:MAP:COUNT? .....	104
7.3.16	:WFR:MAP:CR? .....	105
7.3.17	:WFR:MAP:CSV.....	105
7.3.18	:WFR:MAP:CSV? .....	106
7.3.19	:WFR:MAP:XML.....	107
7.3.20	:WFR:MAP:XML? .....	108
7.3.21	:WFR:MAP:XY .....	109
7.3.22	:WFR:MOV:CR .....	109
7.3.23	:WFR:NEXT? .....	110
7.3.24	:WFR:PATREC .....	110
7.3.25	:WFR:PATREC? .....	111
7.3.26	:WFR:POS:CR? .....	111
7.3.27	:WFR:REFCR Set the Column, Row position.....	111

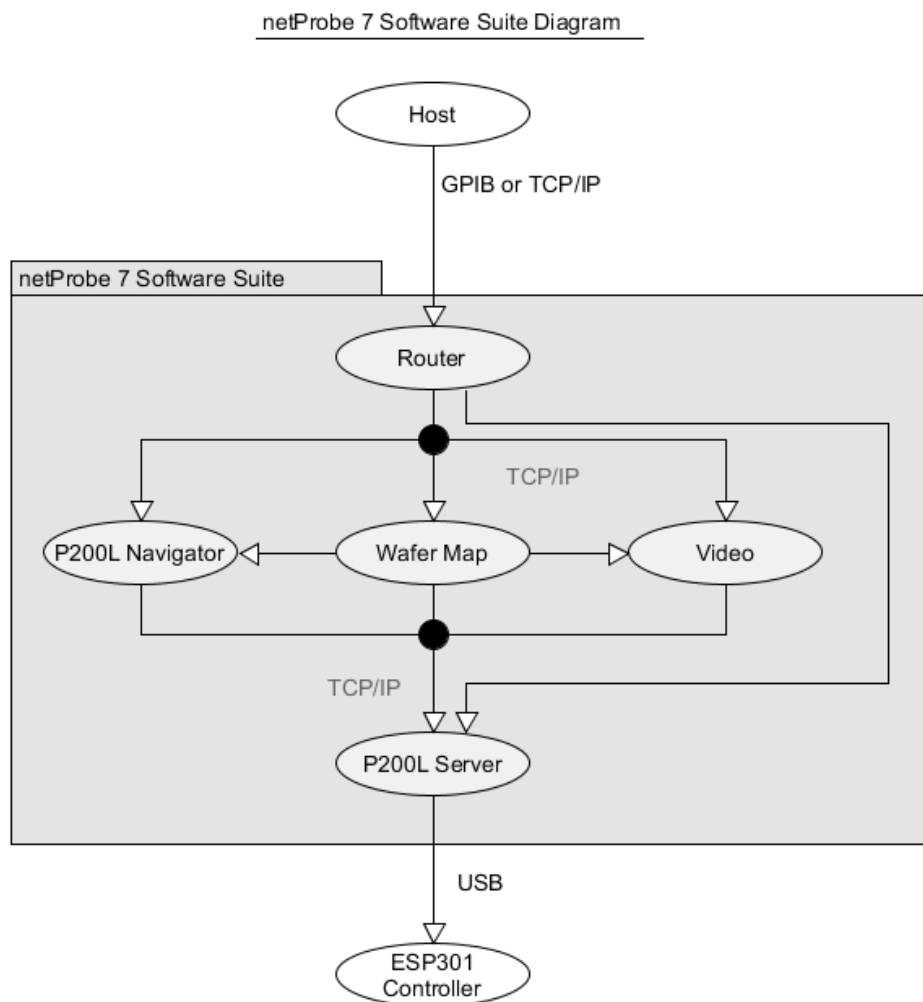
7.3.28	:WFR:REFCR?.....	112
7.3.29	:WFR:RESET.....	112
7.3.30	:WFR:SUBSITES.....	113
7.3.31	:WFR:SUBSITES?.....	113
7.4	Video TCP/IP Commands.....	114
7.4.1	:PAT:CAMERA.....	114
7.4.2	:PAT:CAMERA?.....	114
7.4.3	:PAT:CLEARREF.....	115
7.4.4	:PAT:CORRECT.....	115
7.4.5	:PAT:CORRECT?.....	116
7.4.6	:PAT:METRICS.....	116
7.4.7	:PAT:METRICS?.....	117
7.4.8	:PAT:SHOWREF.....	117
7.4.9	:PAT:TEACH.....	118
7.4.10	:PAT:TEMPLATE.....	118
7.4.11	:PAT:TEMPLATE?.....	119
7.5	Router TCP/IP Commands.....	120
7.5.1	UNITS.....	123
7.5.2	UNITS?.....	123

# 1 netProbe7 Software Suite for P200L

netProbe 7 is the Windows 7/Windows 10 control software for the P200L prober. It provides navigation, imaging, wafer mapping, and external host control of the prober.

**NOTICE: The netProbe 7 computer is a dedicated host for the P200L and P300L probers. Installation of software or hardware on the netProbe 7 computer without the permission of the Micromanipulator Company is prohibited and voids the prober warranty.**

## 1.1 Software components





### 1.1.1 Applications

**netProbe7 P200L Server** – Managed access point for prober control. Multiple applications can communicate to the server using the TCP/IP command set. The GUI applications are launched from the server.

**netProbe7 P200L Navigator** – Basic setup and navigation functions. Includes a subsite list. A command set is available through TCP/IP.

**netProbe7 Video** – Live imaging, mouse navigation, and pattern recognition. A command set is available through TCP/IP.

**netProbe7 Wafer Map** – Wafer navigation and auto-stepping. Connects to Navigator to run subsites at each die, and to Video to get pattern recognition correction. A command set is available through TCP/IP.

**netProbe7 Router** – Managed access point to the netProbe7 applications. Communicates to a host through GPIB and TCP/IP. Uses a TCP/IP command pass-through mechanism which greatly reduces the overhead required to communicate to the netProbe7 applications. If a command is sent attempting to utilize the functionality of an application which is not running, an error is generated.

### 1.1.2 Libraries

**National Instruments VISA 17.5** – Used with the P200L Server (USB to ESP301) and Router (GPIB).

**Micromanipulator\_netProbe7 1.1.2** – Provides shared core functions for all netProbe7 applications. Specifically, command processing and TCP/IP interfacing.

**Emgu CV 3.1.0** – Open source imaging library used by Video

## 1.2 Computer Login

The computer supplied with the P300L is configured with the following login credentials:

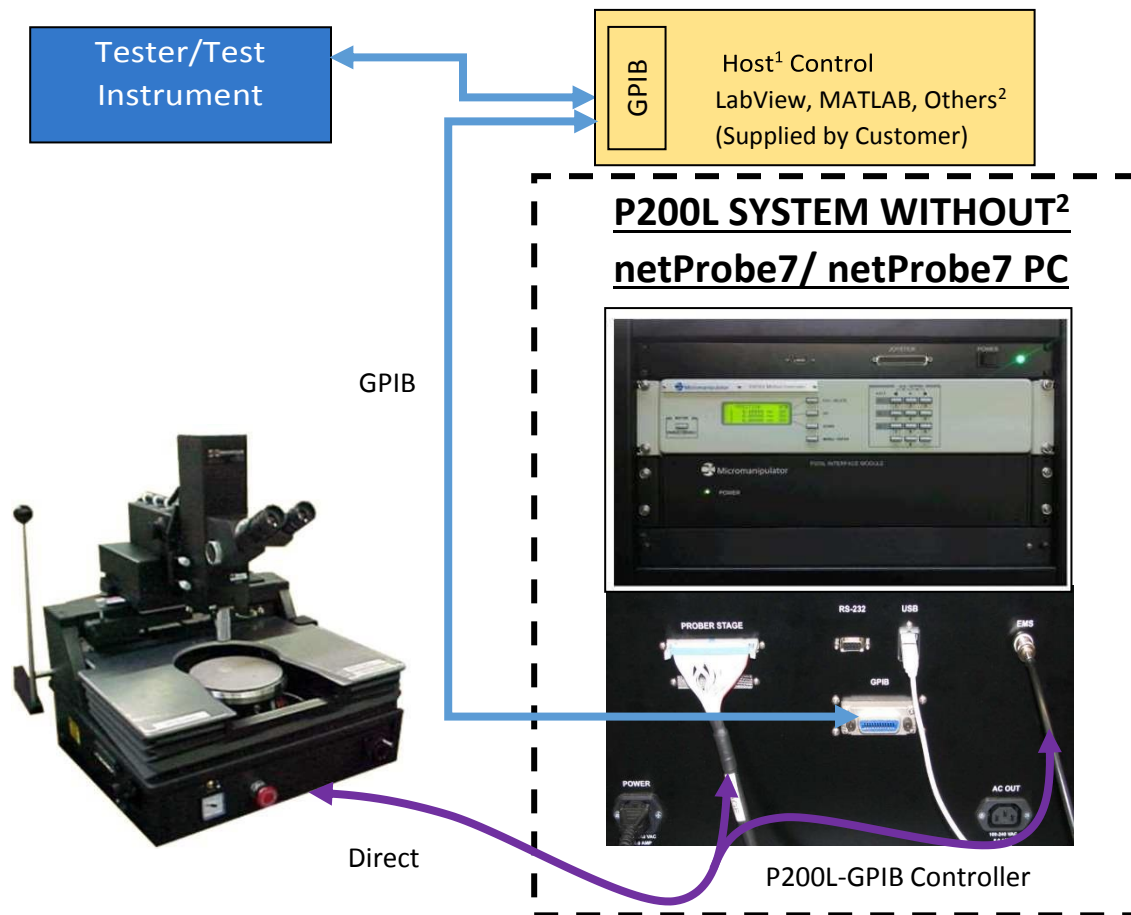
**USER:** Customer

**Password:** Password

### 1.3 GPIB Connections

This section describes the four GPIB prober connection configurations available. The two configurations without netProbe 7 require the P200L-GPIB controller. Configurations with netProbe 7 are compatible with either the P200L-GPIB or P200L-USB controller because netProbe 7 communicates to the P200L controller through USB. **netProbe 7 does NOT communicate to the P200L-GPIB controller through GPIB.** netProbe 7 Router provides the GPIB interface to the host through the National Instruments GPIB-USB-HS adapter.

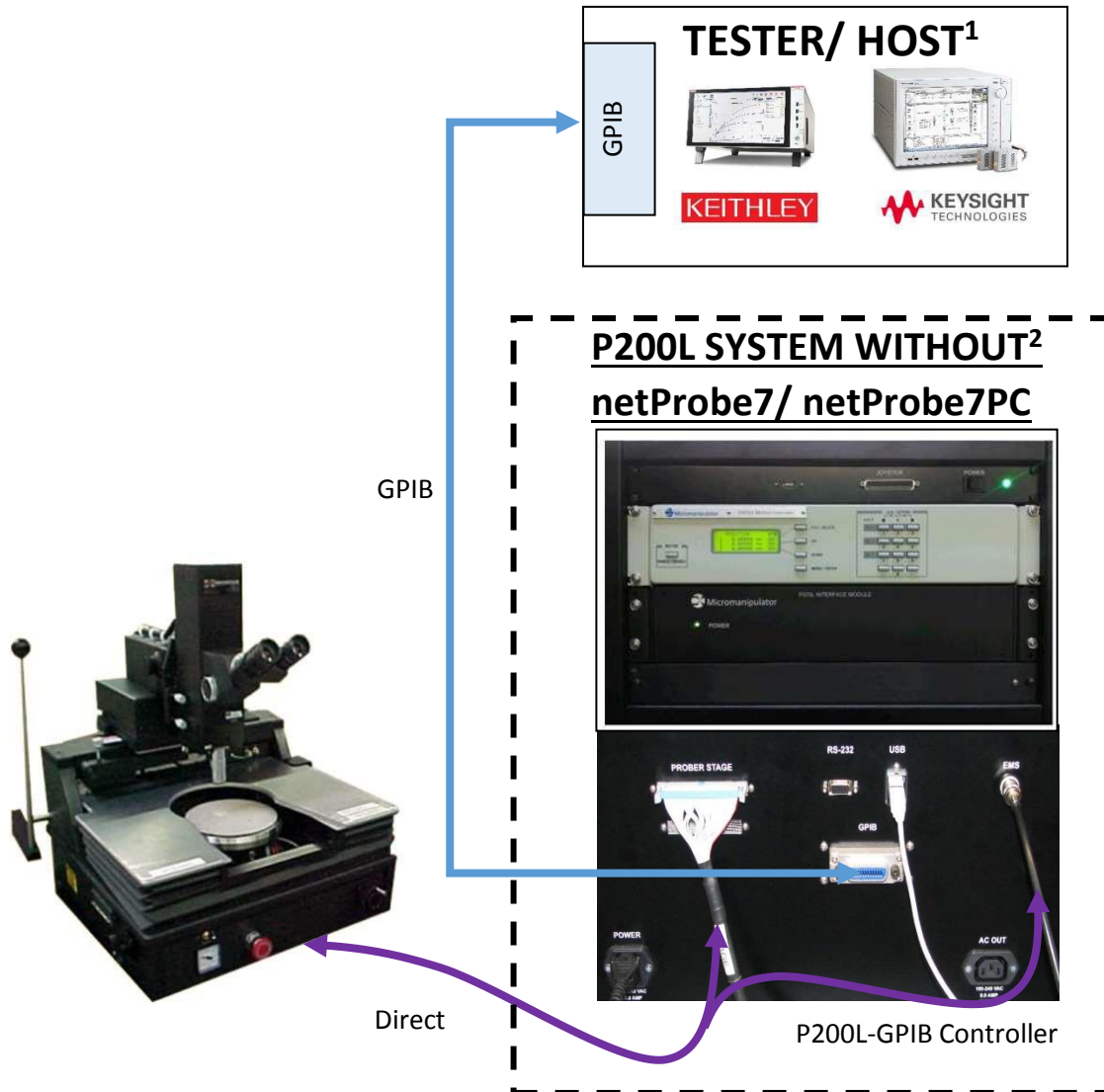
#### 1.3.1 Test Host connected to P200L without netProbe 7 and netProbe 7 PC



<sup>1</sup> HOST is an external PC supplied by the customers / user. A Tester may act as a host if it has control capability.

<sup>2</sup> No netProbe7 programs are available in this configuration. Control is via Joystick and front panel or direct low-level GPIB commands only. Note the P200L controller comes with NI LabView Virtual Instrument Command Library.

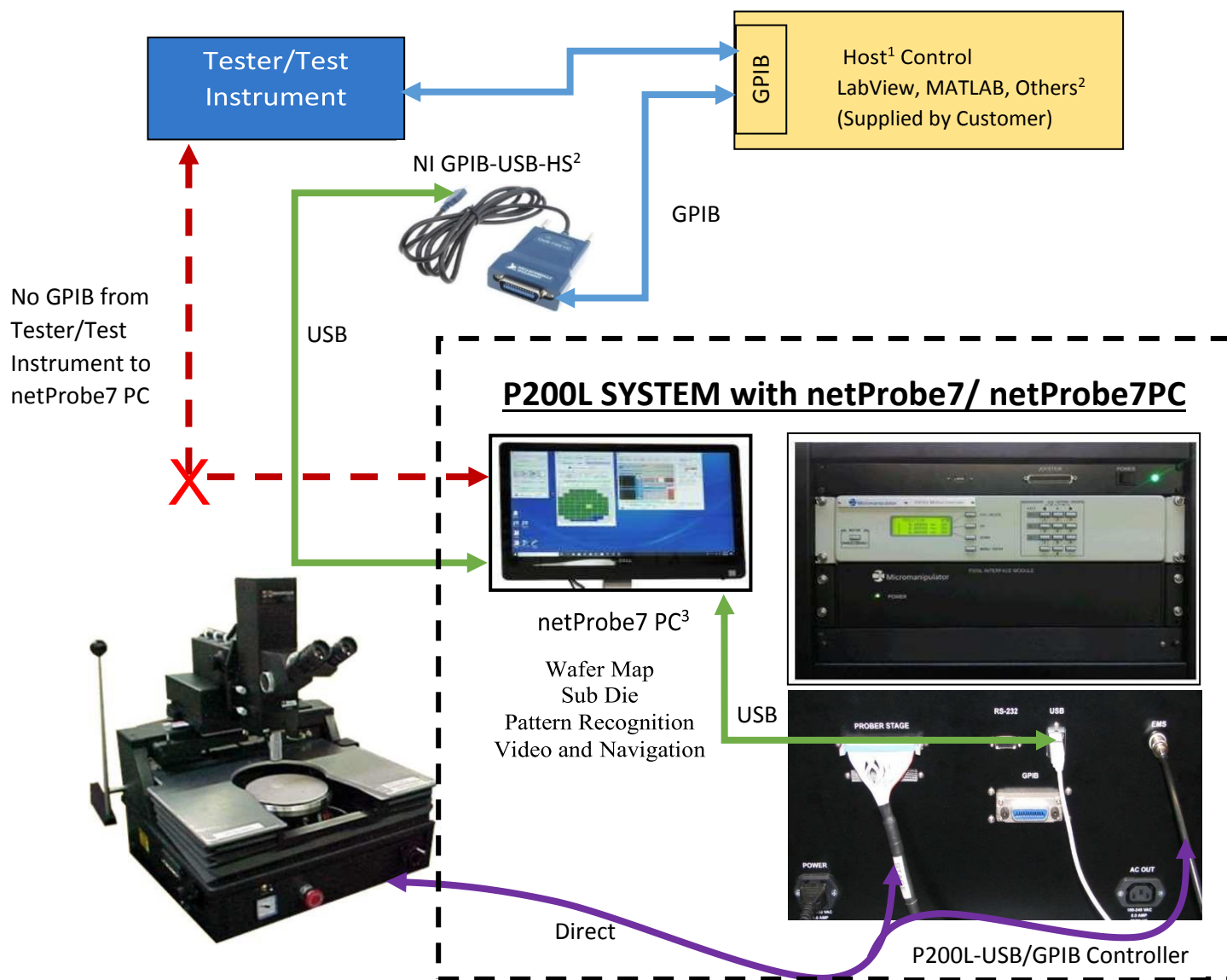
1.3.2 Tester connected to P200L without netProbe 7 and netProbe 7 PC



<sup>1</sup> Tester/ HOST is a Tester / Test Instrument with Host capability supplied by the customers / user.

<sup>2</sup> No NetProbe7 programs are available in this configuration. Control is via Joystick and front panel or direct low-level GPIB commands only. Note the P200L controller comes with NI LabView Virtual Instrument Command Library.

### 1.3.3 Tester/Test Host connected to netProbe 7/netProbe 7 PC

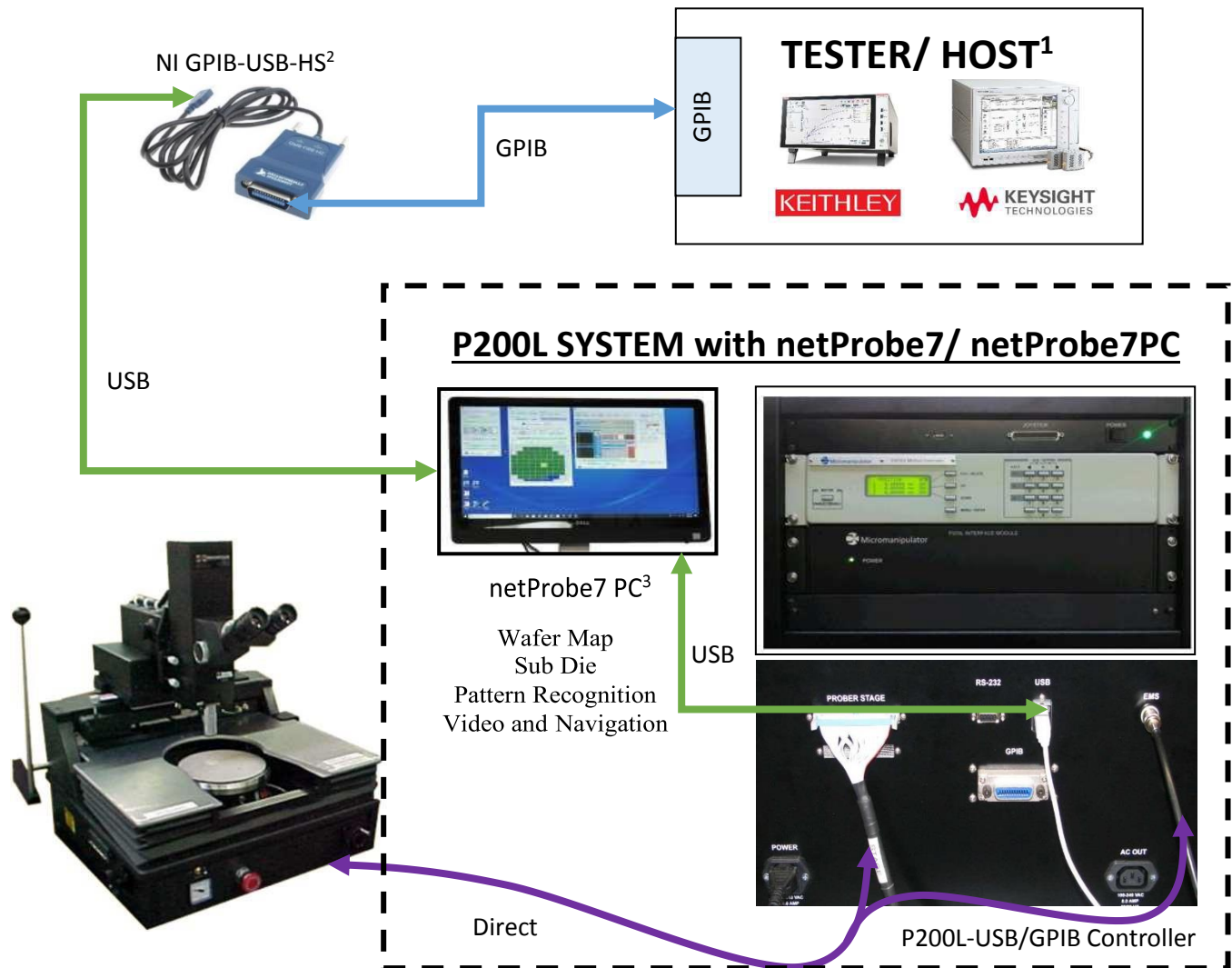


<sup>1</sup> HOST is an external PC supplied by the customers / user. A Tester may act as a host if it has control capability.

<sup>2</sup> This NI GPIB-USB-HS cable / interface is supplied with netProbe7 PC

<sup>3</sup> The NetProbe7 PC is a dedicated CPU for control of the P200L Controller. If you try to use it to run programs to control other instruments those programs may interfere with NetProbe7 operation. Such use of the system is not covered under warranty.

1.3.4 Tester connected to P200L with netProbe 7 and netProbe 7 PC

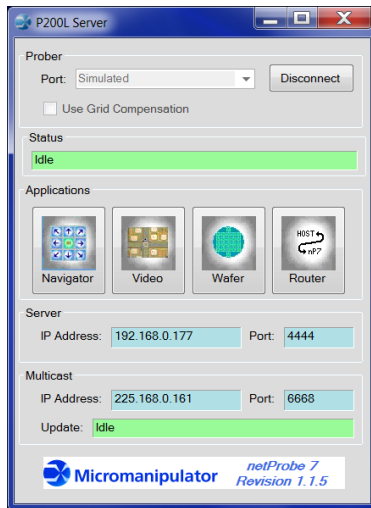


<sup>1</sup> Tester / HOST is a Tester / Test Instrument with Host capability supplied by the customers / user.

<sup>2</sup> This NI GPIB-USB-HS cable / interface is supplied with netProbe7 PC

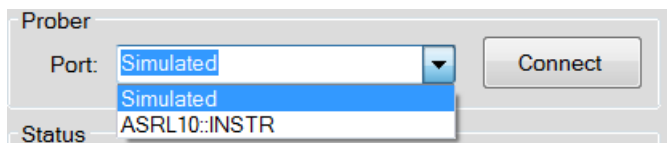
<sup>3</sup> netProbe7 resides on and runs from the NetProbe7 PC. No other programs reside on this computer. If you try to use it to run programs to control other instruments those programs may interfere with netProbe7 operation. Such use of the system is not covered under warranty.

## 2 P200L\_Server

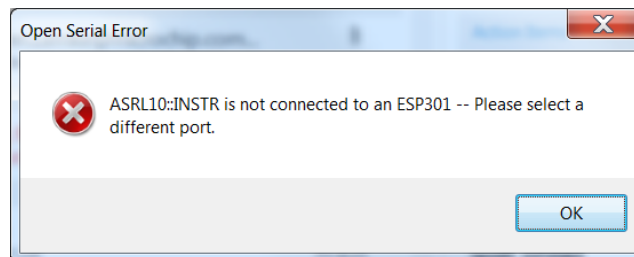


The P200L\_Server provides a single point of access to the ESP301 motion controller. All applications requiring motion control functions must utilize the TCP/IP interface of the server. The server provides a launcher for the Micromanipulator netProbe 7 GUI applications. The server must be running for the GUI applications to access the motion controller.

The **Prober** group contains the prober **Port** selection list and the **Connect/Disconnect** button. The **Port** list always provides the “**Simulated**” option. The simulated mode allows operators to learn the software and developers to test their software without requiring probing hardware.



Select the VISA USB port from the list and then the **Connect** button to start the server. If the port is not connected to an ESP301, the following message is shown:



The **Use Grid Compensation** check box allows the operator to enable or disable the grid table positioning compensation generated at the factory. If the grid table file is not present or is invalid, this option is disabled.

The **Applications** group provides buttons to launch the netProbe 7 GUI applications. The applications do not need to be launched in any particular order, however the Wafer map application requires that Navigator and Video be running if the map test requires the sub-site list and pattern recognition.

The **Server** group displays the IP address and port of the server. All TCP/IP communication to and from the server is through the address and port displayed.

The **Multicast** group contains the IP Address and port of the multicast broadcast. The **Update** field displays the current information being broadcast. Multicasting allows simultaneous transmission of information to multiple network listeners. The implementation runs on top of the User Datagram

Protocol (UDP). Note that all computers and devices connected to the same network as the netProbe7 computer will see the multicast broadcast. The port can be changed if it conflicts with another device on the network. Check with the network administrator for the best settings. The settings to change, as required, are:

```
<MulticastIP>225.168.0.161</MulticastIP>  
<MulticastPort>6668</MulticastPort>
```

The settings are found in the following files:

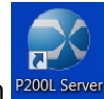
```
C:\netProbe7\bin\P200L_SETTINGS.XML  
C:\netProbe7\bin\ROUTER_SETTINGS.XML  
C:\netProbe7\bin\SERVER_SETTINGS.XML  
C:\netProbe7\bin\VIDEO_SETTINGS.XML  
C:\netProbe7\bin\WAFERMAP_SETTINGS.XML
```

Valid Multicast IP addresses are in the range 224.0.1.0 to 239.255.255.255. Valid port values are 0 to 65535. The settings MUST be the same in all five files. Otherwise, the netProbe7 applications will not update properly.

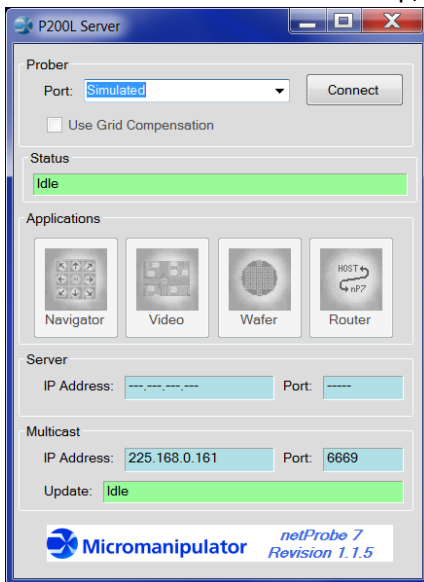
## 2.1 Startup

This is a typical startup sequence for the netProbe 7 server. The screen images show the Prober Port as “Simulated” for documentation purposes.

1. Connect the P200L controller to the computer via the provided USB cable and power them up.
2. When the computer displays the user and password screen for the first time, use the factory default user (Customer) and password (Password).

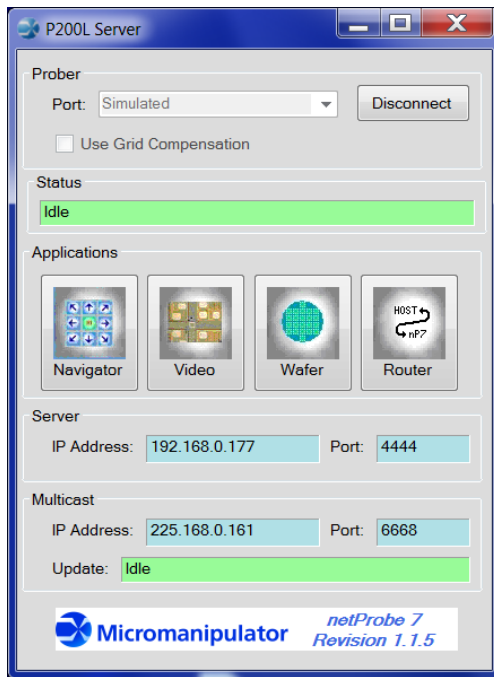


3. The **Server** is started by clicking on the P200L Server Icon on the desktop or on the taskbar. After the server starts up, the screen looks like this:





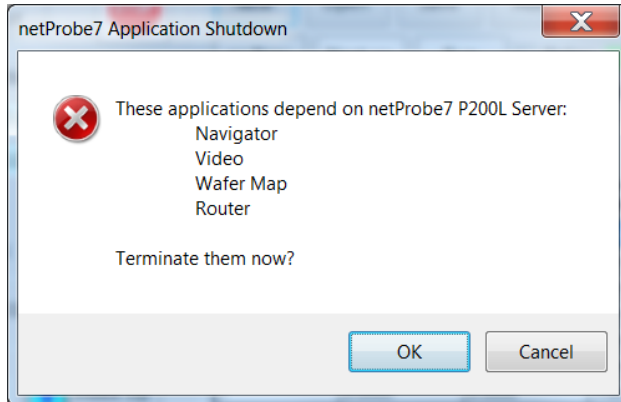
4. Select the USB port connected to the ESP301 from the port list and then select **Connect**. If the connection is successful, the following screen is shown:



5. The Server IP Address and port are filled in and the Applications buttons are now enabled. Applications are started by clicking on the button associated with the application. If the application is already running, it is shown on top of the other open windows.

## 2.2 Shut Down

Select either the **Disconnect** button or the close window button. There is a prompt to ensure the server can be shut down as shown below if an application dependent on the server is running:



If **Cancel** is selected, the server and dependent applications continue running.

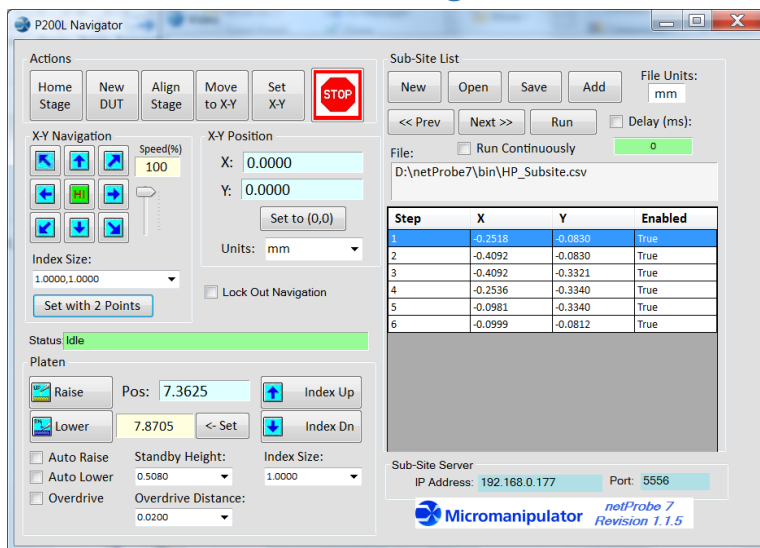
## 2.3 TCP/IP Configuration

TCP/IP communication is through a specific port on the local computer. The IP address and port are displayed in the main window. A client socket can connect to the server, as outlined in this resource:

<https://docs.microsoft.com/en-us/dotnet/framework/network-programming/asynchronous-client-socket-example>

The server requires text commands to be terminated with carriage return (CR). Returned status and data strings are always terminated with CR.

### 3 netProbe7 P200L Navigator



The Navigator application provides prober setup and navigation functions. A Sub-Site list is provided to store and recall locations.

#### Navigation Functions:

- X-Y Jog/XYZ Index
- Raise/Lower Platen
- Move X-Y
- Enable/Disable Operator Navigation

#### Setup Functions:

- Home XYZ
- Load/Unload DUT
- Align DUT
- Set Reference
- Set Down Position
- Set Standby Height
- Auto Z Raise/Lower on Programmed move
- Set Positioning Units

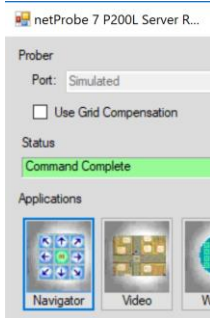
## Sub-Site Functions:


- Add/Delete/Replace/Insert Location
- Enable/Disable Location
- Move to Location
- Move to Next/Previous Location in List
- Open/Save Sub-Site List in either Comma Separated Values(.CSV), or Extensible Markup Language (.XML) format.
- Step through Entire List (Run) with Optional Delay

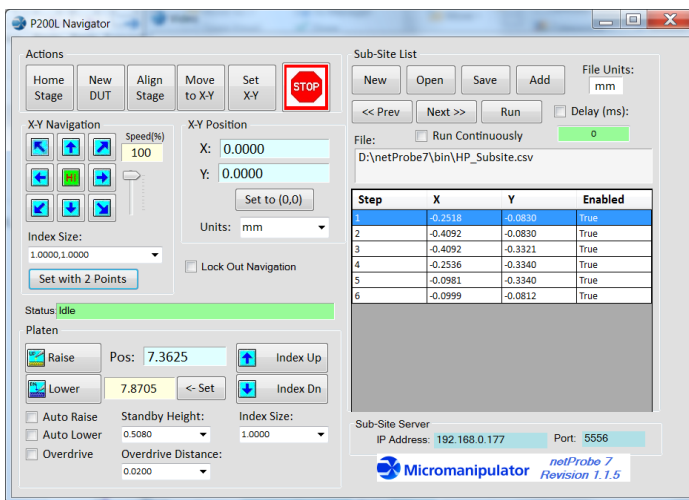
## 3.1 Getting Started

### 3.1.1 Startup

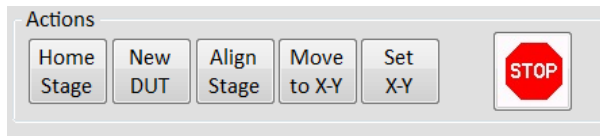
1. Make sure the P200L Server and ESP301 controller are up and running. The P200L Server window is shown as follows:



2. Click on the  button. netProbe 7 P200L Navigator loads and the main screen is displayed as shown below.




### 3.1.2 Recommended Setup Sequence



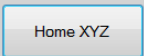
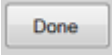
This section describes the steps required to ensure optimum positioning performance of the prober. The essential operations described in this section are:


- **Home Stage (see 3.1.3)** -- Establish the XYZ axes reference for the prober. This operation is usually not required unless the system was not properly shut down.
- **New DUT (see 3.1.4)** – Load a wafer on the chuck.
- **Align Stage (see 3.1.5)** -- Establish the rotational orientation of the wafer relative to the stage.
- **Z Setup (see 3.1.6)** – Set the probe tip contact position and standby height

### 3.1.3 Home Positioners


1. Select the  button in the “Actions” group. The following dialog box is shown:



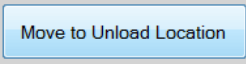
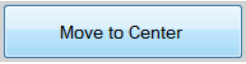
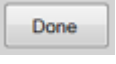
2. Ensure that probes will safely clear any obstacles or objectives when the platen is homed.
3. Select  button to move the platen and stage to their respective home positions. The platen raises to the top limit. The stage moves to the left rear corner of travel.
5. Click  when all moves are completed.

**NOTE:** If the  button is pressed during a home operation the embedded controller resets the HOMED state. This disables all programmable XYZ functions. Home must be successfully completed to regain programmable XYZ functions.

### 3.1.4 Loading a Wafer or DUT

1. Select the  button from the “Actions” group. The New DUT dialog box appears as shown below.




2. Select  to move the stage to the preset unload location.
3. Load a new DUT on the chuck.
4. Select  to move the prober stage to its center of travel.
5. Select  to exit the dialog box.

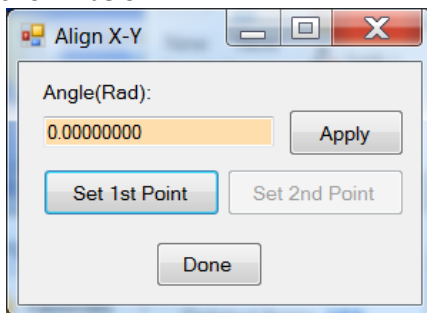
### 3.1.5 Aligning X-Y Stage

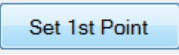
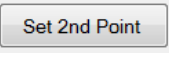

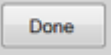
The alignment of the stage is required to ensure accurate positioning to wafer locations. The best alignment method is to mechanically align the wafer X axis to the stage X axis using the manual theta adjustment knob.

If it is not possible to mechanically align the chuck due to test fixture constraints, it is necessary to determine the angle between the X stage and X wafer axes mathematically and use that angle when calculating programmed moves.

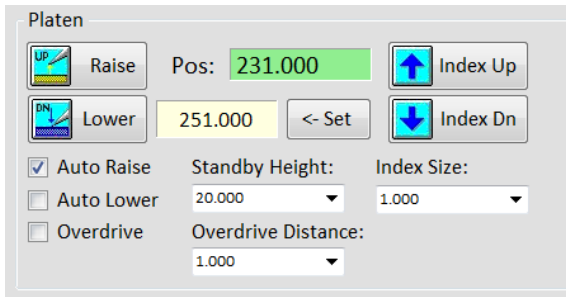
1. Focus on the wafer surface and center the Video crosshair on a horizontal feature.



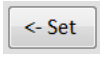
2. Select the  button in "Actions" group. The Align X-Y dialog box is displayed as shown below.



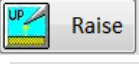

3. Position the probe stage such that the video crosshair is perfectly aligned with the bottom edge of a die on the middle left side of the wafer. Be sure to approach the point from the upper right to ensure accuracy.
4. Click on the  button to set the first alignment location.
5. Move the probe along the row of die to the right until the video crosshair is perfectly aligned with the bottom edge of a die on the right side of the wafer.
6. Click on .
7. Click on . The angle of the DUT is now synchronized with the XY stage.
8. Test the alignment by indexing in X along a horizontal feature. The horizontal crosshair remains aligned with the feature when moving along the wafer when alignment is set correctly.
9. Perform steps 3 through 8 until the required alignment accuracy is attained.
10. Press  to close the dialog box.

### 3.1.6 Z Setup

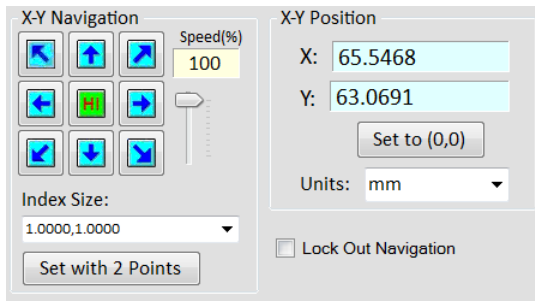


1. Turn off the **Auto Raise** and **Auto Lower** options. Set **Standby Height** to 20 mils (508 microns), and an **Index Size** of 1 mil (25.4 microns)  

2. Rotate the X, Y, and Z knobs on each manual manipulator until the X,Y, and Z stages are near the center of available manipulator travel. This gives the broadest range of axis adjustment.
3. Observe the Z probe tip positions on all the manipulators. If the Z height of all the probe tips vary by more than 100 mils (2540 microns) it may be necessary to adjust the probe tips in the probe holders until all the tips are close to the same height.
4. Identify the manipulator with the lowest probe tip. If there is some doubt, lower one until it is clearly the lowest.
5. Place the manipulators approximately in the final configuration.
6. Observe the Z distance between the probe tips and the surface of the wafer.
7. Lower the platen using either the joystick or index down button until the lowest probe tip is about 20 mils (500 microns) above the wafer. When using the joystick, as the probe tips approach the wafer select low speed mode to prevent unintentional probe contact with the wafer.
8. Select the  button.
9. Place the lowest manipulator such that the tip is over its intended target.
10. Select the lowest power objective (usually 2X).
11. Move the microscope over the manipulator from step 9.
12. Select the next highest objective and adjust the microscope position until the target is visible.
13. Slowly lower the platen until the probe tip contacts the target. Larger tipped probes usually start to scrub across the wafer when contact is made. Very small tipped probes tend to penetrate the surface. It may be necessary to adjust the X-Y position of the manipulator tip before contact is made to ensure that the probe tip contacts the target.
14. Select the  button to set the probe contact position.



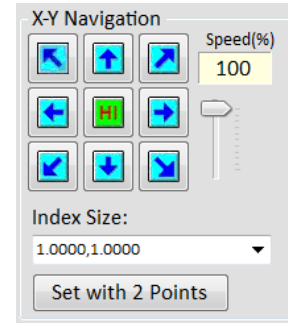
15. Select the **Platen Raise** button  . The platen should raise by the standby distance.
16. Select the **Platen Lower** button  and verify probe contact.
17. Move the microscope to the next pad, center the crosshair, and then position the next manipulator such that the probe tip is in the field of view.
18. Adjust the Z position of the probe tip until it makes contact.
19. Repeat steps 17 and 18 until all probes are making contact.

## 3.2 Coordinate Positioning



### 3.2.1 X-Y Indexing with Navigation Jog Buttons

1. Enter the desired X-Y index size in the X-Y index box.
2. Click on the LO/HI/IDX button until IDX is selected as shown at right.
3. Click and the left mouse on the jog button pointing in the required direction on the wafer or DUT. The index move is performed when the jog button is released.

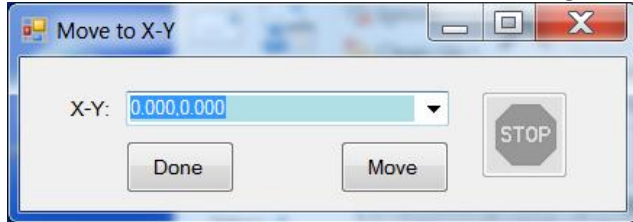


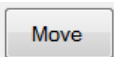
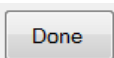
NOTE: An emergency stop button is available in the **Actions** group. This button causes the current move to be immediately stopped.



### 3.2.2 Directed X-Y Moves

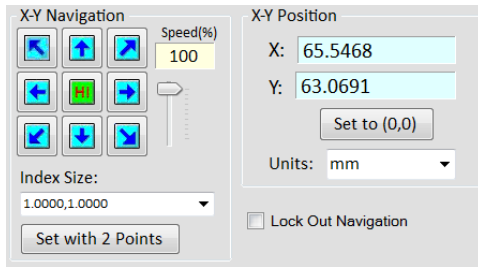
1. Select the  button. The Move X-Y dialog is as shown below.



2. Select a position from the drop-down list or enter the X-Y destination value.
3. Select  to move the stage to the specified destination. The **Stop** button is enabled during the move in case the move needs to be immediately halted.
4. Select  when the directed move is complete.

### 3.3 Navigator Reference

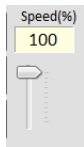
#### 3.3.1 X-Y Stage



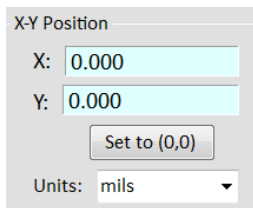
The navigation keys provide a full range of control for both direction and speed of the stage. In **LO** and **HI** jog modes, the stage motion is continuous in the direction of travel selected. In **IDX** (Index) mode, the positioner moves by the index distance in the selected direction.

Repeated selection of the HI button steps through the following jog mods:

- HI – High speed continuous jog
- LO – Low speed continuous jog mode calculated as 1/10th HI speed
- IDX – Index mode which moves the probe by the index size at high speed when a jog key is pressed.



The speed slider provides control of the continuous jog speed in HI/LO jog modes. It has a range of 0.1% to 100% of peak stage speed. The speed slider does NOT change the speed of programmed XYZ moves.



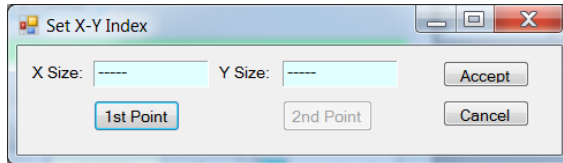
The X-Y position of the active positioner is continuously updated using the selected units. This position is referenced to a user defined position, set with

either the **Set to (0,0)** button or through the **Set X-Y** dialog box. Available units are millimeters (mm), microns ( $\mu\text{m}$ ), and 1/1000<sup>th</sup> inch (mils). The active units are used for every position related display in Navigator, including the Sub-Site List.

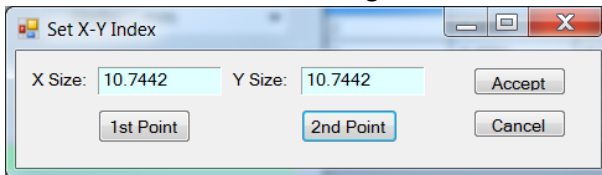
### 3.3.2 Set X-Y Index Using Two Points

The index size can be set by calculating the distance between two points using the Set X-Y Index dialog box.

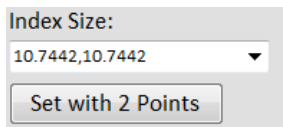
1. Select the **Set with 2 Points** button. The Set X-Y Index dialog box is shown.



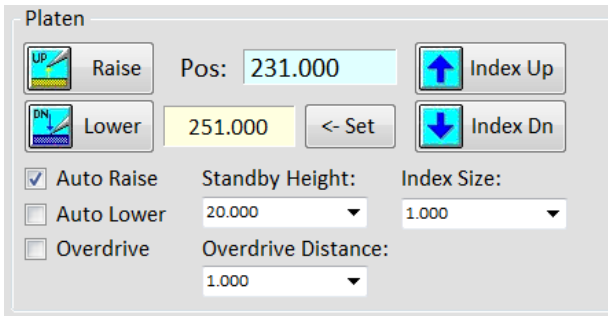
2. Position the stage until the Video crosshair is over the first point and select **1st Point**.
3. Position the stage until the crosshair is over the second point and select **2nd Point**. The X and Y size is calculated and the dialog box looks like this:



4. Select **Accept**. The dialog box closes and the **Index Size** is updated to the new value as shown below.

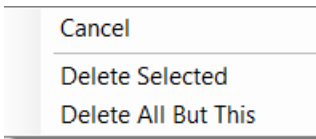


### 3.3.3 Z Platen



The Z position of the platen is displayed using the selected units. The Z Down set position is displayed below the platen position. The standby height and overdrive distance are shown beneath the down set position number. The background color for the position display changes based on the position of the platen relative to the down set position according to the following table:

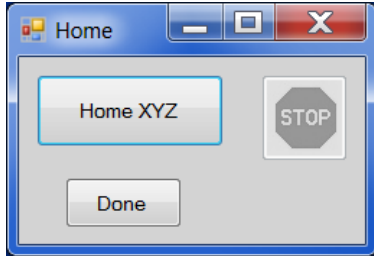
Color	Description
Green	Platen is up
Yellow	Platen is at down position
Red	Platen is below down position



The Z settings combo box list contents can be cleaned up by right-clicking in the text entry field. A pop-up menu is shown. The **Delete Selected** option deletes the entry in the text box. The **Delete All But This** option deletes the entire list except for the entry in the text box.

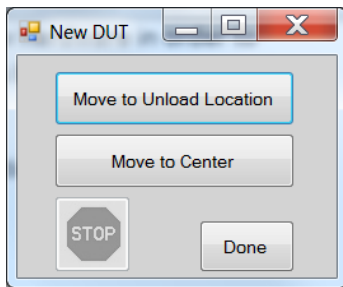
### 3.4 Actions

#### 3.4.1 Home



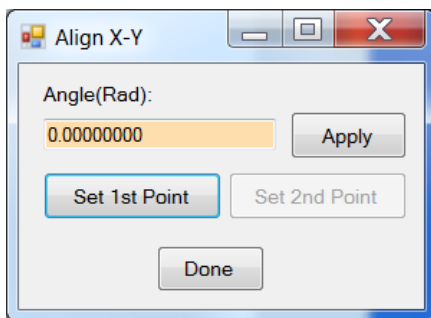
The probe is homed using the Home dialog box. The Stop button is enabled during the homing operation in case there is a problem. The Stop button sends a “:PRB:KI” command to the server, immediately terminating the home move.

#### 3.4.2 New DUT



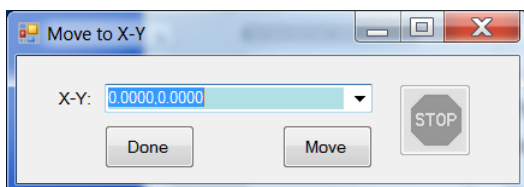
The probe stage is moved to the front of travel to load a new wafer. The stage is then moved back to center to resume probing. The Stop button is enabled during the move.

#### 3.4.3 Align Stage



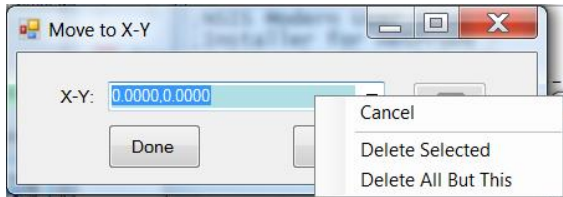
The alignment of the X-Y axes of the stage with the X-Y axes of the DUT is accomplished using two point alignment. Since the P200L does not have a programmable theta, the alignment angle is used by the server whenever an X-Y move is performed, or position is requested.

#### 3.4.4 Move to X-Y



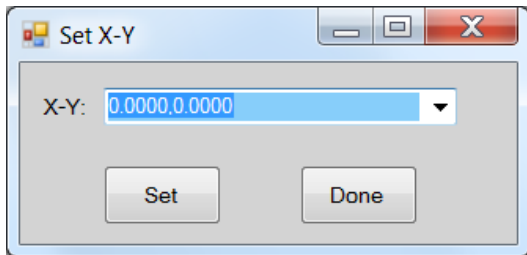
X-Y moves can be performed by entering the X-Y value. The values entered are saved in the combo box list for re-use. The Stop button is active only during moves.

The combo box list contents can be cleaned up by right-clicking in the text entry field. A pop-up menu is shown as follows:



The **Delete Selected** option deletes the entry in the text box. The **Delete All But This** option deletes the entire list except for the entry in the text box.

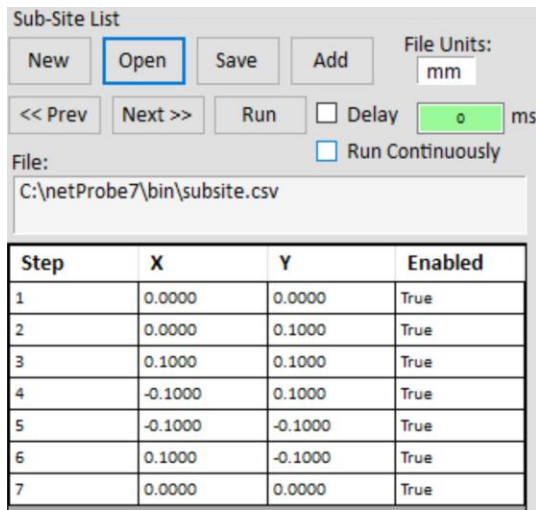
### 3.4.5 Set X-Y



The position of the prober can be defined using the Set X-Y dialog box. The values entered define the new user reference for X-Y positions. The values are saved in the combo box for re-use. The contents of the list can be cleaned up by right-clicking in the text entry field as described in section 3.4.4.



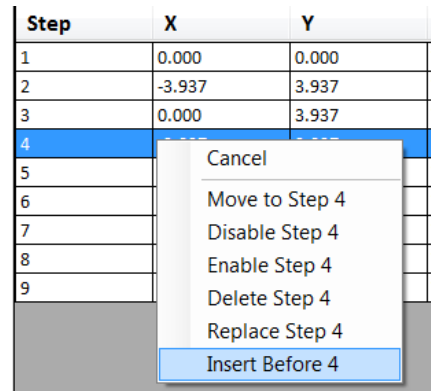
### 3.4.6 Sub-Site List



The Sub-Site List provides a list of stored X-Y stage locations. The locations are always referenced to the current user defined (0,0). The list is loaded by opening a file with the **Open** dialog box. The list is **Run** as a sequence with an optional delay between steps. The list can be run continuously. The **Prev** and **Next** buttons cause a move to the previous or next enabled location. If any modifications are made, the file can be saved by pressing the **Save** button. Files are saved in either Excel CSV format or in XML format.

A new list is created using the **New** button. Locations may be added to the end of the active list by pressing the **Add** button. The units used by Navigator when the list is created are stored in the file. There is no maximum established for the number of steps allowed.

Right clicking on a selected row displays the popup menu at right. The **Move to Step** selection causes the defined positioner to move to the location. The **Disable Step** selection disables a step so that it is ignored during single stepping and running. The location can be re-enabled by selecting the **Enable Step** selection. The **Delete Step** selection removes the step from the list. The **Replace Step** selection stores the X-Y position of the stage in the selected step. The **Insert Before Step** selection inserts a step with the current location before the selected step.



### 3.4.6.1 Sub-Site List File Format

#### **Excel CSV:**

```
type=subsite
units=mm | μm | mils
step1,x1,y1,enabled1
.
.
.
stepn,xn,yn,enabledn
```

Where *step* is the sequential step number

*x,y* is the XY location

*enabled* indicates if the step is enabled (False | True)

For Example,

```
type=subsite
units=mm
1,0.0000,0.0000,False
2,0.0000,0.100,True
3,0.1000,0.100,True
4,-0.1000,0.100,True
5,-0.1000,-0.100,True
6,0.1000,-0.1000,True
7,0.0000,0.0000,True
```

#### **XML:**

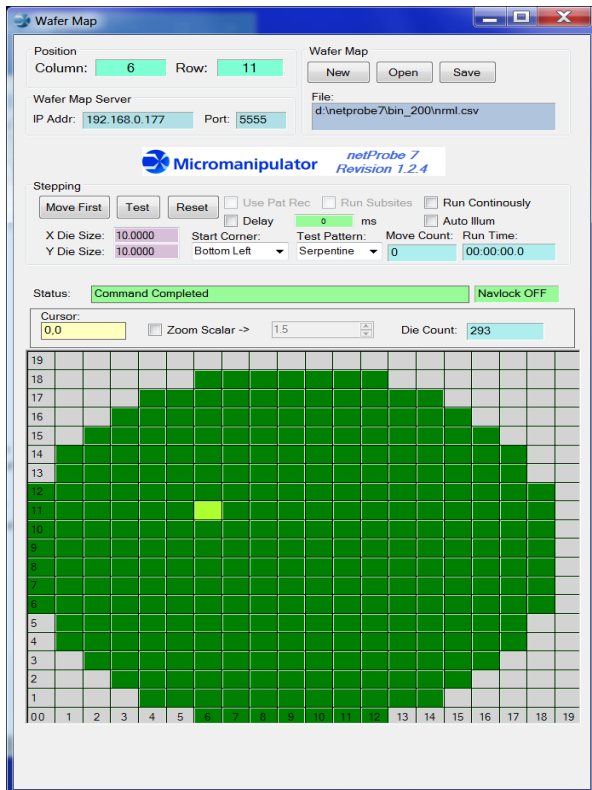
```
<?xml version="1.0" encoding="utf-8"?>
<SUBSITES>
  <UNITS>mm | μm | mils </UNITS>
  <STEP>
    <ID>step1</ID>
    <X>x1</X>
    <Y>y1</Y>
    <ENABLED>enabled1</ENABLED>
  </STEP>
  .
  .
  .
  <STEP>
    <ID>stepn</ID>
    <X>xn</X>
    <Y>yn</Y>
    <ENABLED>enabledn</ENABLED>
  </STEP>
</SUBSITES>
```

Where *step* is the sequential step number  
*x,y* is the XY location  
*enabled* indicates if the step is enabled (False | True)

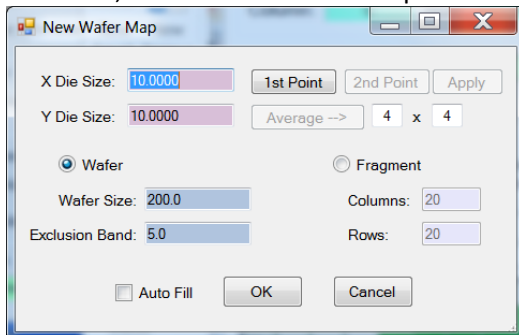
For Example,

```
<?xml version="1.0" encoding="utf-8"?>
<SUBSITES>
  <UNITS>mm</UNITS>
  <STEP>
    <ID>1</ID>
    <X>0.0000</X>
    <Y>0.0000</Y>
    <ENABLED>False</ENABLED>
  </STEP>
  <STEP>
    <ID>2</ID>
    <X>0.0000</X>
    <Y>0.1000</Y>
    <ENABLED>True</ENABLED>
  </STEP>
  <STEP>
    <ID>3</ID>
    <X>0.1000</X>
    <Y>0.1000</Y>
    <ENABLED>True</ENABLED>
  </STEP>
  <STEP>
    <ID>4</ID>
    <X>-0.1000</X>
    <Y>0.1000</Y>
    <ENABLED>True</ENABLED>
  </STEP>
  <STEP>
    <ID>5</ID>
    <X>0.1000</X>
    <Y>-0.1000</Y>
    <ENABLED>True</ENABLED>
  </STEP>
  <STEP>
    <ID>6</ID>
    <X>0.0000</X>
    <Y>0.0000</Y>
    <ENABLED>True</ENABLED>
  </STEP>
</SUBSITES>
```

## 4 netProbe7 Wafer Map



The Wafer map program provides a representation of the die to test on a wafer or wafer fragment. The map can be used for Column, Row navigation and auto-stepping. It connects to Navigator to make use of Sub-Sites, and to Video to access pattern recognition.



Wafer maps are created by selecting the “New” button. The die size can be entered or calculated using two points.

It is possible to create a map for wafer fragments.

Die can be added, removed, and disabled directly on the map using the mouse.

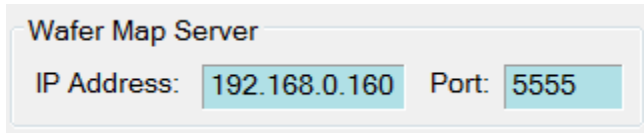
## 4.1 Overview

### 4.1.1 Wafer Map Window



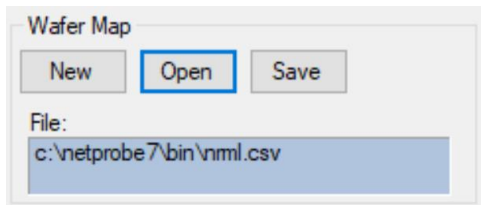
Position  
Column: 8 Row: 1

The Column and Row position is calculated from the XY die size entered. The reference for the die map is set from the map popup menu.



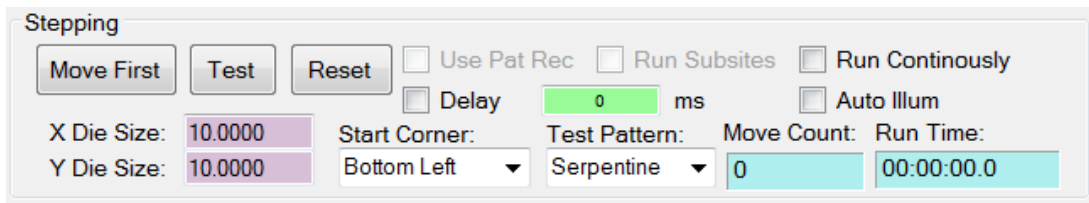
Wafer Map Server  
IP Address: 192.168.0.160 Port: 5555

The Wafer Map Server accepts TCP/IP commands from external programs at the displayed address and port.



Wafer Map  
New Open Save  
File:  
c:\netprobe7\bin\nml.csv

Wafer maps can be saved and opened in either CSV or XML format (see section 4.3 for the file formats). The active file is shown in the “File:” text box.



Stepping  
Move First Test Reset  Use Pat Rec  Run Subsites  Run Continuously  
 Delay 0 ms  Auto Illum  
X Die Size: 10.0000 Start Corner: Bottom Left Test Pattern: Serpentine Move Count: 0 Run Time: 00:00:00.0  
Y Die Size: 10.0000

Wafer stepping is performed using the options available in the **Stepping** group. The **Move First** button moves the stage to the first testable die and resets the tested state of all the die on the map. The **Test** button executes a test of all the testable die on the wafer. The **Reset** button resets the tested state of all the die on the map.

The **Use Pat Rec** option is enabled if Video is running. When checked, every CR move is corrected to match the pattern recognition template. This ensures nearly perfect positioning performance.

The **Run Subsites** option causes a wafer test to move to all the steps in the Sub-Site list at each die of the wafer.

The **Run Continuously** option causes a wafer test to repeat endlessly.

The **Delay** option causes the wafer test to pause by the entered milliseconds at each die.

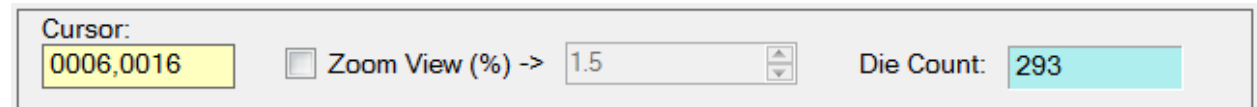
The **Auto Illum** option causes the illuminator to turn on before a die move, and turn off afterward.

The **X Die Size** and **Y Die Size** text boxes allow entry of a different die size for the active map. This value is NOT saved in the map file.

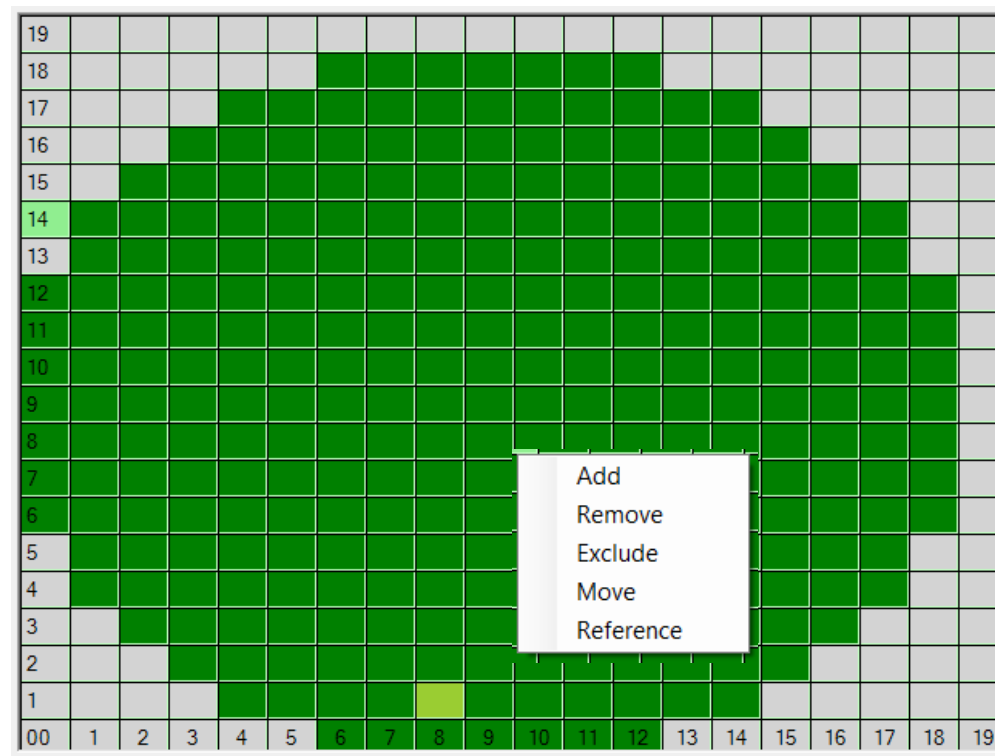
The **Start Corner** selection defines the corner with the first die of the wafer test. The **Test Pattern** selection defines the way the stage moves to the next row. The **Serpentine** option causes the stage to move to the die on the end of next row down closest to the current die. The **ZigZag** pattern always starts a row test on the same side as the start corner.



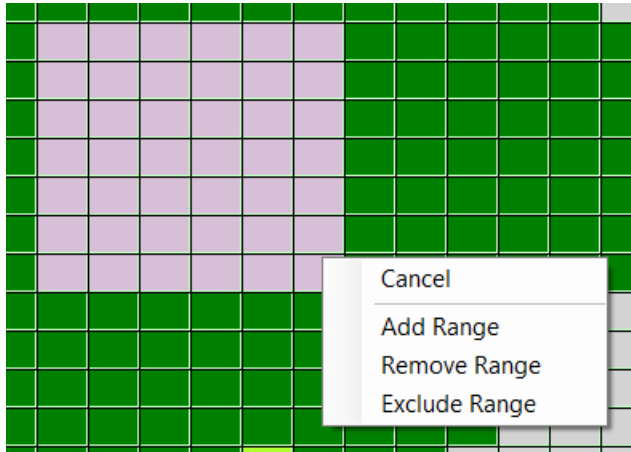
The **Status** text box shows the current server command and relevant test status messages. The **Navlock OFF/ON** text box shows the status of the joystick navlock state.



The **Cursor** text box shows the location of the mouse cursor on the wafer map. The **Zoom View** option enables increasing or decreasing the size of the displayed wafer map. The map automatically maintains focus on the current die location. The **Die Count** is the number of testable die on the wafer.

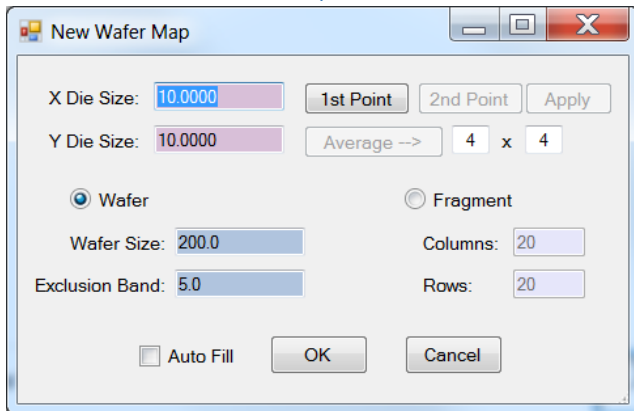


The right-click popup menu provides map editing and navigation functions. The functions require that a die is selected by Left-Clicking on a die. The **Add** function adds the die to the testable die list. The **Remove** function removes the die from the testable die list. The **Exclude** function marks that die as untestable. The **Move** function moves the stage to that die. The **Reference** function forces the CR position to be the same as the selected die.



There is a popup menu which appears when multiple die are selected by pressing Shift while Dragging the mouse. The **Add Range** function adds all the selected die to the testable list. The **Remove Range** function removes all the die from the testable list. The **Exclude Range** function marks the selected die as untestable.

## 4.2 Create a new map



### 4.2.1 Set Die Size

The XY die size can be entered manually or calculated from two points (**Apply**). The die size must be correct to ensure accurate stepping. The most accurate method is to measure across several die, and average the distance by the number of die between the two points.

#### Averaging Method

1. Navigate to a die at the bottom left of the wafer.

2. Position the Video crosshair on an easily locatable artifact on the die. Use the Point-and-Shoot mouse function of Video to more accurately position the video crosshair over the artifact.
3. Select **1st Point**.
4. Navigate up and over at least 4 die.
5. Adjust the position of the prober until the crosshair is perfectly aligned with the artifact from step 2.
6. Select **2nd point**.
7. Enter the number of columns and rows moved in the two data fields found after the **Average→** button.
8. Select **Average -->**. This calculates the average distance between the die. The calculated die size is placed in the Die Size boxes.

#### 4.2.2 Selecting a map type

Two types of maps are available – **Wafer** and **Fragment**. The Wafer map is a circular map of die calculated from the wafer size, die size, and exclusion band. The **Exclusion Band** is the distance from the edge of the wafer consisting of ugly or partial die. The Fragment map is a rectangular array of die arranged Columns wide by Rows high.

#### 4.2.3 Generating the Map

A map is created by selecting **OK** to close the dialog box. If the **Auto Fill** option is checked, the map is filled with testable die. If this option is not checked, the map is empty.

### 4.3 Wafer Map File Format

#### Excel CSV:

```
SubstrateType,Wafer
Units,units
WaferSize,diameter
Rows,rows
Columns,cols
DeviceSizeX,xdie
DeviceSizeY,ydie
ExclusionBand,bandsize
NullBin,0
Bin,G
Bin,X
Bin,I
Orientation,0
OriginLocation,origin
Spline,spline
Row 0, bc00...bc0n
.
.
.
Row m, bcm0...bcmn
```

Where **units** is mm | micron | mils



**diameter** is the wafer diameter  
**rows** is the total number of rows  
**cols** is the total number of columns  
**xdie** is the column size using the defined units  
**ydie** is the row size using the defined units  
**bandsize** is the distance from the perimeter containing only ugly or edge die  
**origin** is the starting corner with 0=lower left 1=lower right 2=upper right 3=upper left  
**spline** is the pattern used for testing with 0=serpentine and 1=zigzag  
**Row m** defines a row of die  
**bcmn** is a bincode matching a [row,col] where bincode is one of the following:  
0=Untestable G=Passed X=Failed I=Ignore T=Untested

For Example,

```
SubstrateType,Wafer
Units,mm
WaferSize,150.0000
Rows,8
Columns,7
DeviceSizeX,20.2132
DeviceSizeY,19.8769
ExclusionBand,0.0000
NullBin,0
Bin,G
Bin,X
Bin,I
Orientation,0
OriginLocation,0
Spline,0
Row 0,IIGGGII
Row 1,ITTGGGI
Row 2,TTTTTTT
Row 3,TTTTTTT
Row 4,TTTTTTT
Row 5,ITTTTTI
Row 6,IITTTII
Row 7,IIIIIII
```

## XML:

The XML format conforms to the Sematech definition for wafer map files.

```
<?xml version="1.0" encoding="utf-8"?>
<Maps>
  <Map SubstrateType="Wafer">
    <Device Rows="rows" Columns="cols" NullBin="0" WaferSize="diameter"
DeviceSizeX="xdie" DeviceSizeY="ydie" Units="units" Orientation="0"
OriginLocation="origin">
      <ReferenceDevice ReferenceDeviceX="refx" ReferenceDeviceY="refy" />
      <Data>
        <Row><![CDATA[bc00...bc0n]]></Row>
        .
        .
        .
        <Row><![CDATA[bcm0...bcmn]]></Row>
      </Data>
    </Device>
  </Map>
</Maps>
```

Where **units** is mm | micron | mils

**diameter** is the wafer diameter

**rows** is the total number of rows

**cols** is the total number of columns

**xdie** is the column size in *microns*

**ydie** is the row size in *microns*

**origin** is the starting corner with 0=lower left 1=lower right 2=upper right 3=upper left

**spline** is the pattern used for testing with 0=serpentine and 1=zigzag

**Row m** defines a row of die

**bcmn** is a bincode matching a [row,col] where bincode is one of the following:

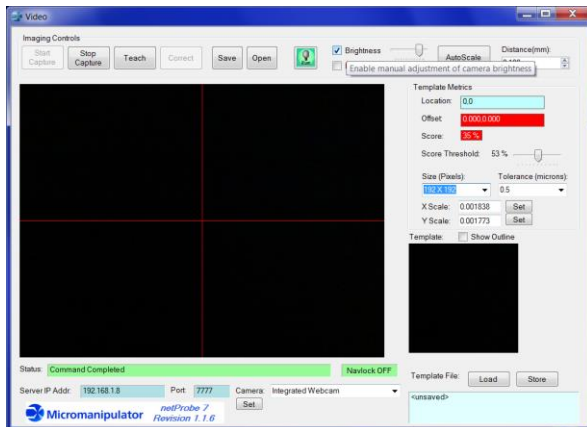
0=Untestable G=Passed X=Failed I=Ignore T=Untested

For example,

```
<?xml version="1.0" encoding="utf-8"?>
<Maps>
  <Map SubstrateType="Wafer">
    <Device Rows="8" Columns="7" NullBin="0" WaferSize="150.0000"
DeviceSizeX="20213.2000" DeviceSizeY="19876.9000" Units="mm" Orientation="0"
OriginLocation="0">
      <ReferenceDevice ReferenceDeviceX="10" ReferenceDeviceY="10" />
      <Data>
        <Row><![CDATA[IIGGGII]]></Row>
        <Row><![CDATA[ITTGGGI]]></Row>
        <Row><![CDATA[TTTTTTT]]></Row>
        <Row><![CDATA[TTTTTTT]]></Row>
      </Data>
    </Device>
  </Map>
</Maps>
```

```
<Row><![CDATA[TTTTTTT]]></Row>
<Row><![CDATA[ITTTTTI]]></Row>
<Row><![CDATA[IITTTII]]></Row>
<Row><![CDATA[IIIIIII]]></Row>
</Data>
</Device>
</Map>
</Maps>
```

## 5 netProbe7 Video

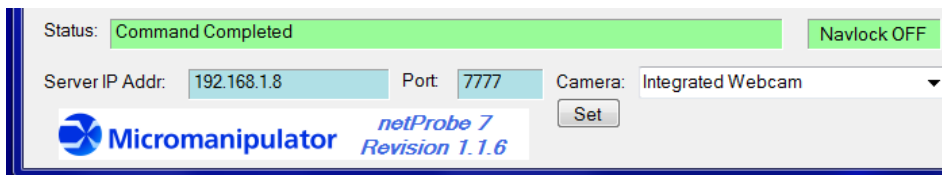


The netProbe 7 Video application provides live video and pattern recognition. It includes a resizable floating video window with the same capabilities as the main video window, e.g. click-and-drag/point-and-shoot navigation, teach template, and move to corrected position. Templates can be stored and loaded to allow re-use with a particular set of wafers. If the necessary hardware for illuminator control is connected, the illuminator can be turned on and off with the **Illum** button.

### 5.1 Startup

#### 5.1.1 Camera Selection

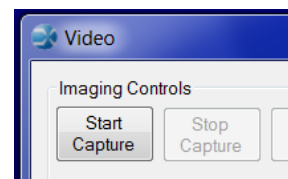
There is usually only a single camera connected to the netProbe 7 computer, so the camera is selected automatically. If there are multiple cameras, select the camera mounted on the prober microscope. For example, the *StUSBCam* option in the camera list below is the prober camera. Select that camera from the list and then select “**Set**” to use it.



#### 5.1.2 Live Video

Live video is started by selecting the **Start Capture** button.

Select the **Stop Capture** button to freeze the video on the current frame.



### 5.1.3 Setting up Pattern Recognition

The pattern recognition functions require an image template and scalars correlating a pixel in the image to the physical XY movement of the stage.

#### 5.1.3.1 Establish XY Scaling

##### Automated

1. Move the stage until a unique feature is centered under the crosshair.
2. Select a template size of 128x128.
3. Select the **Teach** button.
4. Select an **Autoscale** distance which will keep the feature in the field of view when moved.
5. Select **Autoscale**. The stage moves by the distance and calculates the correct XY scale values.

##### Manual

1. Using the joystick or Navigator, move the stage until a unique feature is centered under the crosshair.
2. Set the current XY position to (0,0) by selecting the **Set to (0,0)** button in *Navigator*.
3. Set the XY scale values to 1.0.
4. Select a template size of 128x128.
5. Select the **Teach** button.
6. Move the stage diagonally until the unique feature is half way to the corner of the image.
7. Read the XY position shown on Navigator and write it down.
8. Read the XY offset in Video and write it down.
9. Calculate the XY scale values by dividing the XY position by the XY offset.
10. Enter the calculated XY values.

### 5.1.3.2 Teach a Template

A template is used as the reference image for pattern recognition. The features within the template are located in the live image. The position of the features in the live image are compared to the position of those same features in the template. This positional offset of the features between the images is used to perform position adjustments.

Templates can be taught by selecting the **Teach** button or with the mouse by pressing **Ctrl** and **Left** clicking the mouse. The **Teach** button teaches a template which uses the center of the live image as the template reference. The **Location** text box in the **Template Metrics** group. The **Ctrl Left** click method uses the position of the mouse cursor in the image as the template reference. Note that only templates taught with the **Teach** button can be stored. This limitation is in place because by default the center of the loaded template image is used as the center of the feature of interest.

The features in the template must be unique in the image, otherwise the search will be incorrect.

1. Select the template size. The size should be large enough to contain the feature in the image.
2. Select the score threshold. The most consistent results are obtained with a threshold of around 50%.
3. Either select the **Teach** button or position the mouse cursor over the center of the feature and **Ctrl Left** click.

### 5.1.4 Functions using Mouse/Key Combinations

#### Click-and-Drag

The Click-and-Drag function moves the stage by the relative distance displayed at mouse cursor in the video image.

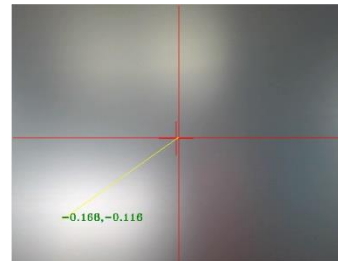
1. Move the mouse cursor over a feature to move.
2. Press the **LEFT** mouse button.
3. Move the mouse cursor in the image to the desired position for the feature.
4. Release the **LEFT** mouse button.
5. The stage moves by the indicated distance.



#### Point-and-Shoot

The Point-and-Shoot function moves the stage to the location selected on the video image.

1. Move the mouse cursor in the image to the desired position for the taught pattern.
2. Click the **RIGHT** mouse button.
3. The stage moves to the selected position.



#### Perform Correction

Correction moves use the offset between the template and the live image to ensure that the stage is positioned precisely.

1. Move the mouse over the live image.
2. Press the **Ctrl** key and click the **RIGHT** mouse button.
3. The stage corrects by the measured offset.

#### Teach Template

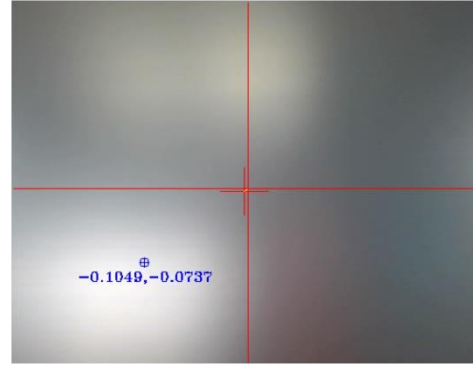
Templates are used to establish a reference image for use by pattern recognition. The features within the template are located in the live image, and the resulting offset is used to perform position adjustments.

1. Move the mouse cursor over the feature to use as a template.
2. Press the **Ctrl** key and click the **LEFT** mouse button.
3. The new template is taught. The active template is displayed to the right of the live image.

#### Teach Reference Mark

The Reference mark can be used to indicate a feature of interest in an image. Its position is relative to the center of the taught template. The reference mark is very useful when positioning probe tips on pads as follows:

1. Position a probe over a pad and lower it until it makes good contact.
2. Move the mouse cursor to the tip of the probe in the image and press the **Alt** key and click the **LEFT** mouse button.
3. A **Blue** center mark and position are shown at the end of the probe tip.
4. The reference mark can now be used during navigation to indicate the position of the probe tip at the contact position even if the probe is out of focus.

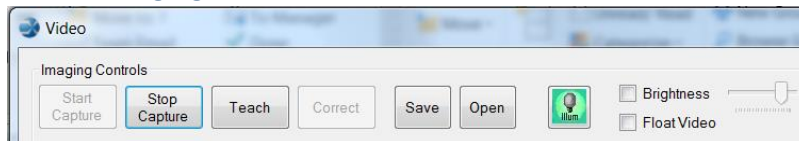


### Clear Reference Mark

1. Move the mouse over the live image.
2. Press the **Shift** key and click the **LEFT** mouse button.
3. The reference mark is cleared.

## 5.2 Reference

### 5.2.1 Imaging Controls



#### 5.2.1.1 Start Capture

**Start Capture** starts streaming the live image from the camera. Pattern recognition is available only when the live image is streaming.

#### 5.2.1.2 Stop Capture

**Stop Capture** freezes the live image on its current frame.

#### 5.2.1.3 Teach

Selecting **Teach** creates a template for use with pattern recognition. Templates are only valid for the microscope magnification and zoom at the time the template was taught. Changing magnification and zoom requires that the template be retaught.

Pattern recognition works best when these template selection guidelines are followed :

- The illumination level on the sample should be bright when viewed through the microscope. This minimizes the effects of color and contrast variation.
- The defined template should contain a well defined (high contrast) light/dark boundary -- the edge of a die, for instance. This minimizes the effects of focus and illumination variations.
- The light/dark boundary in the template must occur once in the entire viewable image.



- The template cannot depend on fine details for uniqueness. Details get mangled and lost when an image is slightly out of focus, causing the pattern recognition function to fail. For this reason, image enhancement add-ons should not be used with pattern recognition.
- Do not include stationary probes in the template since they do not move with the image.
- The template should be defined using features found at EVERY die.
- Templates are valid only for the scope zoom and magnification settings at the time the model is defined.
- Templates should be towards the center of the field of view. This maximizes the likelihood of the template being in the field of view on another area of the wafer when doing alignment and correction.

#### 5.2.1.4 *Correct*

Selecting **Correct** causes a correction move to occur when a template is found in the live image. The distance to move is calculated by multiplying the pixel offset between a feature in the live image and the same feature in the template by the XY scalars.

#### 5.2.1.5 *Save*

**Save** freezes the image and opens a Save As dialog box. The displayed image is saved as a JPG, BMP, or PNG file. Live imaging resumes after the dialog box is closed.

#### 5.2.1.6 *Open*

The **Open** button opens a dialog box which opens a JPG, BMP, or PNG file for viewing. Live imaging is stopped when the selected image is loaded. The **Start Capture** button must be pressed to resume live imaging.

#### 5.2.1.7 *Illuminator Control*

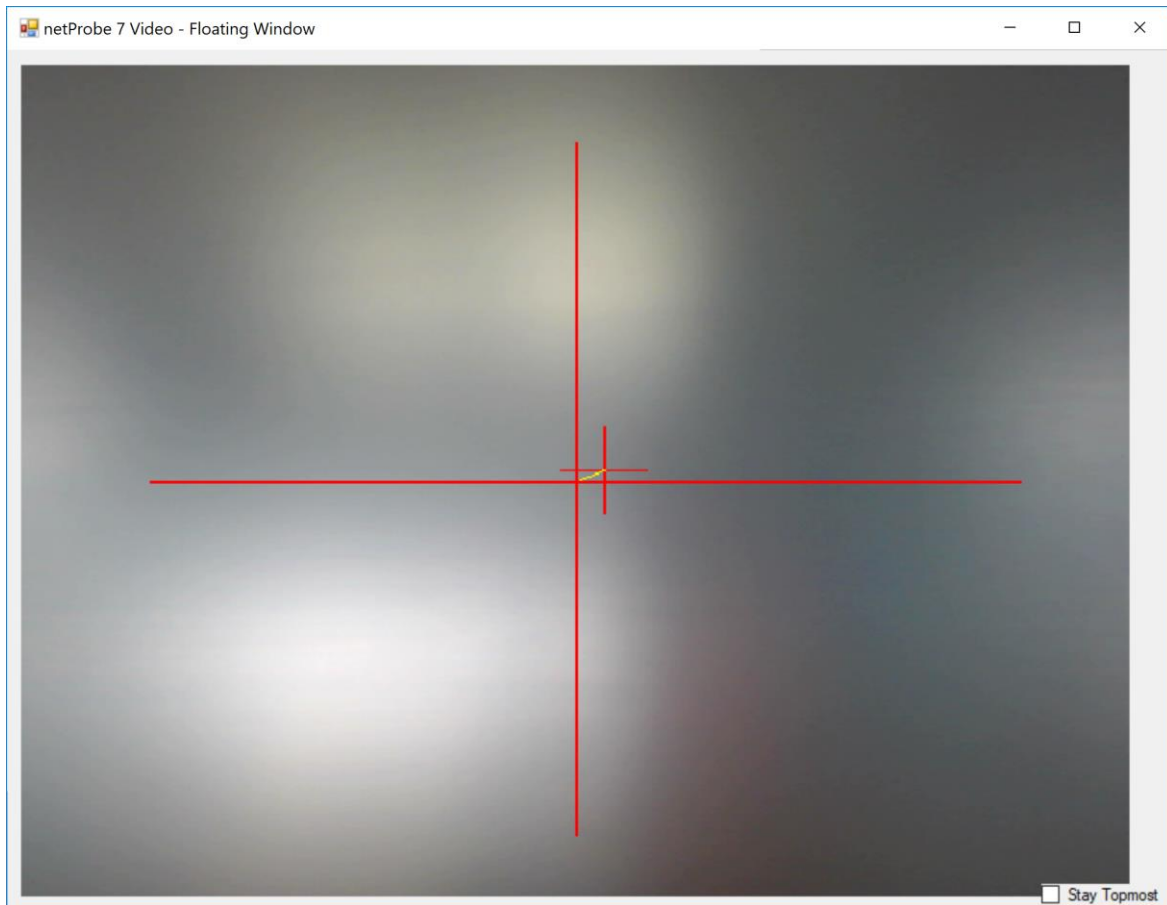
The **Illum** button turns the illuminator on and off. The Illuminator Controller is required to use this button.

#### 5.2.1.8 *Brightness*

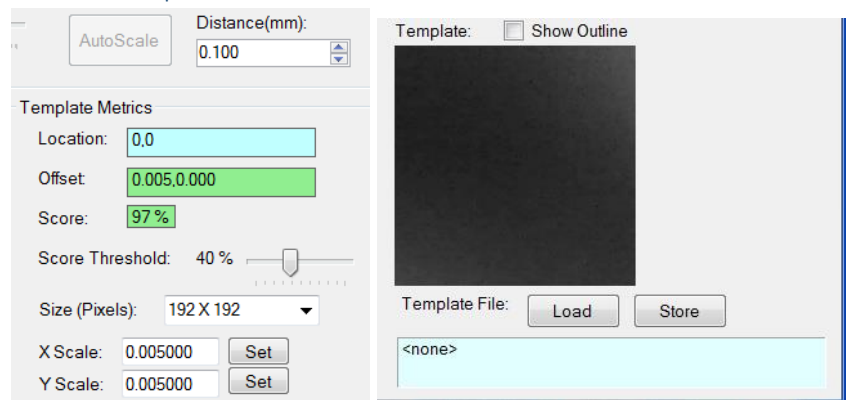
The Brightness check box and slider control are used to adjust the live image brightness. If the check box is unchecked the brightness is set to an optimized value.

### 5.2.1.9 *Float Video*

Selecting **Float Video** opens a floating video window that can be resized and repositioned on the desktop. The checkbox must be unchecked to close the window. It is a duplicate of the live image shown in the main application window. The keyboard and mouse functions are the same as those used with the main window image. The **Stay Topmost** checkbox makes the floating window always stay on top of other application windows.



## 5.2.2 Templates



### 5.2.2.1 AutoScale

Selecting **AutoScale** uses pattern recognition to correlate a known stage movement with pixel displacement in the image. The values calculated are then saved as the XY scalars. The distance moved must keep the reference feature within the field of view in order for pattern recognition to work.

### 5.2.2.2 Location

The **Location** display element is the location of the center of the template within the live image, referenced to the center of the live image.

### 5.2.2.3 Offset

The **Offset** is the distance between a feature in the live image and the same feature in the template. The offset is used when performing correction moves. The background color is **GREEN** if the template is matched in the image, and **RED** if no match is found.

### 5.2.2.4 Score

The **Score** is the pattern recognition match score between the template and the live image. The score must be greater than or equal to the score threshold for pattern recognition for a match to be valid. The background color is **GREEN** if the template is matched in the image, and **RED** if no match is found.

### 5.2.2.5 Score Threshold

The **Score Threshold** defines the **Match/No Match** threshold for pattern recognition. A value of 50% is recommended for most applications. Setting the threshold too low can result in incorrect pattern recognition matches. Setting the threshold too high can result in no match when there is a match.

### 5.2.2.6 Size (Pixels)

**Size** defines the size, in pixels, of the template. The primary constraint for choosing a template size is whether the unique feature fits in the template. Pattern recognition is quicker for smaller templates, but the tradeoff is that the template may not contain enough information for a good match.

### 5.2.2.7 XY Scale

The X and Y Scale values are used to correlate stage motion with display pixels. For example, if XY stage motion of 0.1 mm moves a feature in the image by 32 pixels in X, and 30 pixels in Y, the scalars are

calculated as  $X \text{ Scale} = .1/32 = 0.003125$ ,  $Y \text{ Scale} = .1/30 = .003333$ . These scalars are used when performing correction moves and when using the mouse for navigation on the image.

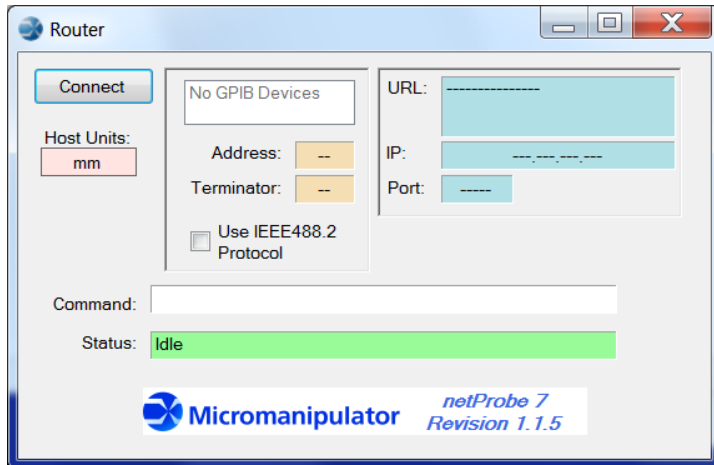
#### *5.2.2.8 Show Outline*

The **Show Outline** checkbox enables or disables the display of the template outline in the image.

#### *5.2.2.9 Load/Store*

The template can be loaded from and stored to a file. This allows for re-use of templates with DUTs, and ensures that pattern matching is consistent across a batch of DUTs.

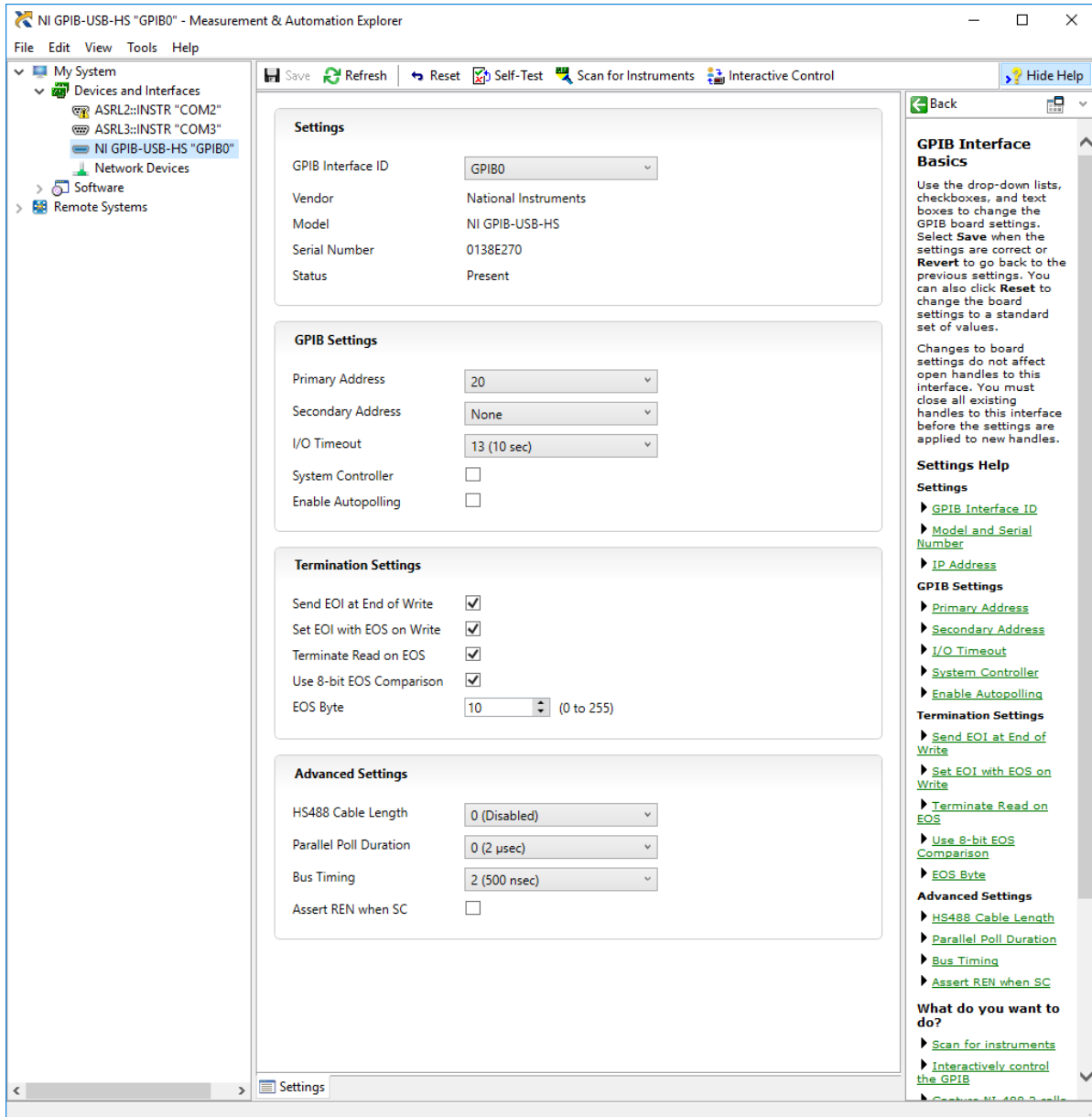
## 6 netProbe 7 Router



The Router application provides a single point of communication between a host computer and the netProbe 7 environment. Router is just that: It redirects commands to the applications which execute the commands. The TCP/IP interface to all the applications uses units of mils. Router converts position related data to the units requested by the test host. The available interfaces are GPIB and TCP/IP. Router has access to all the TCP/IP command sets of the netProbe 7 GUI applications. Note that the asynchronous UDP update functionality of the P200L Server is not available to the host.

**NOTE: The only supported GPIB interface connected to the P300L computer is the National Instruments NI GPIB-USB-HS using National Instruments GPIB/VISA driver version 17.5.**

Router uses the interface settings defined in the NI MAX application on the netProbe 7 computer. For example:



**NOTE:** The test host must either enable Auto Serial Polling in the interface setup, or execute a Serial Poll after each Write or Read operation.

The TCP/IP communications protocol for Router is the same as for the P200L Server. The GPIB interface uses the Serial Poll (SPOLL) byte to report status, rather than the #xxx status format of TCP/IP. For example, the following sequence demonstrates how a move command is handled with GPIB:

GPIB Host	netProbe 7 Router
PRB:MOV:XYA 100 200<LF> →	Line feed terminated command
Serial poll consisting of Request Service (RQS=64) and BUSY (16)	← SPOLL=80 (0x50)
	Prober moves to 100.0000,200.0000
Serial poll consisting of RQS and Data Available (128)	← SPOLL=192 (0xC0)
New XYZ position is returned	← 100.0000 200.0000 35.1000<LF>
Command is complete	← SPOLL=64 (0x40)

When updates are enabled, the following sequence occurs when an operator uses the joystick:

GPIB Host	netProbe 7 Router
UPDT 273<LF> →	Enable updates for joystick state changes and position (0x0111)
Serial poll consisting of Request Service (RQS=64) and BUSY (16)	← SPOLL=80 (0x50)
Command is complete	← SPOLL=64 (0x40)
	Operator actuates X axis on joystick
Serial poll consisting of Data Available (128), Request Service (RQS=64), and Update (8)	← SPOLL=200 (0xC8)
	← :JOY:JOG=true false false<LF>
	X axis begins move
	SPOLL=200 (0xC8)
	← :PRB:POS=103.113 200.000 35.000<LF>
	X axis continues move

	S POLL=200 (0xC8)
	← :PRB:POS=121.987 200.000 35.000<LF>
	Operator releases joystick
	S POLL=200 (0xC8)
	← :JOY:JOG=false false false<LF>
	X axis stops
	S POLL=200 (0xC8)
	← :PRB:POS=133.426 200.000 35.000<LF>



## 7 Application Command Sets

### 7.1 P200L Server TCP/IP Commands

There is one unit of measurement available between the server and GUI applications – mils (1/1000<sup>th</sup> Inch). One significant feature of the command set implementation is that information related to the results of many of the commands is immediately returned. This greatly increases the command processing efficiency since it is no longer necessary to query the server for information after executing a command. For instance, after moving the stage to a location, the actual position of the stage is returned. A typical TCP/IP command sequence looks like this:

GUI Application	P200L_Server
:PRB:MOV:XYA 100 200<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
	Prober moves to 100.0000,200.0000
Data is available	← #192<CR>
New XYZ position is returned	← 100.0000 200.0000 35.1000<CR>
Command is complete	← #064<CR>

#### 7.1.1 Host Command Response Codes

The controller responds to host commands with codes indicating the state of command execution. Control codes are prepended with the “#” character.

Response	Description
#080	Command is accepted and active
#192	Data is available to read
#064	Command successfully completed
#066	Command error
#079	Command aborted

### 7.1.2 Asynchronous Update Format

The server has a UDP multicast channel dedicated to sending data updates. The advantage of this approach is that it is very fast and can broadcast simultaneously to all listeners. This means that all GUI applications are synchronized with the current status of the prober without having to constantly ask for the status. The format of the updates is described below.

The host is notified of asynchronous updates when a state on the controller changes. Updates are used primarily with the knob and joystick. Updates are enabled using the “**UPDT <updt\_field>**” command. The parameter sent with the update command is a bit field integer mapped as follows:

Bit	Decimal (Hex) Value	Description
0	1 (0x0001)	Enable Updates
3	8 (0x0008)	Joystick Navlock (:JOY:NAVLOCK=true false)
4	16 (0x0010)	Joystick Jogging (:JOY:JOG=true false true false true false)
8	256 (0x0100)	Device position before, during, and after joystick jog (:PRB:POS=<x> <y> <z>)

For example, sending “UPDT 9” enables updates whenever the state of Navlock changes.

The general form for asynchronous updates is:

:<device>:<property>=<value1> [value2>...]

Update	Cause	Description
:JOY:IL=<state>	Illuminator state changed	Illuminator on state. e.g. :JOY:IL=true
:JOY:NAVLOCK=<enabled>	:JOY:NAVLOCK command	Navigation lock state. e.g. :JOY:NAVLOCK=true
:JOY:JOG=<x> <y> <z>	Operator actuates or releases the joystick	Send the actuated state of the XY and Z joystick controls when they change. e.g. :JOY:JOG=true false false
:PRB:DNM=<autoup> <autodown> <autooverdrive>	:PRB:DNM command	The auto raise, auto lower, and auto overdrive flags. e.g. :PRB:DNM=true false false
:PRB:POS=<x> <y> <z>	XY programmed move, XY jog, Motion stopped or aborted	Compensated prober XYZ position referenced to the user defined (0,0) offset. e.g. :PRB:POS=100.0000 200.0000 32.3333
:PRB:POSNO=<xno> <yno> <z>	XY programmed move, XY jog, Motion stopped or aborted	Compensated prober XYZ position without user offset. e.g. :PRB:POSNO=-3813.122 2912.544 32.3333
:PRB:UDO=<up> <down> <standby> <overdrive>	:PRB:UDO command	Up and down Z positions, standby height, and overdrive distance. e.g. :PRB:UDO=32.3333 42.3333 10.0000 0.0000
:PRB:X=<x>	:PRB:CEN:X command	Uncompensated (raw) X position. e.g. :PRB:X=-3812.923
:PRB:Y=<y>	:PRB:CEN:Y command	Uncompensated (raw) Y position. e.g. :PRB:Y=2912.311
:PRB:Z=<z>	Z programmed move, Z jog, Raise, Lower	Z position. e.g. :PRB:Z=32.3333
:SYS:OWN=<owner_id> <owned>	:SYS:OWN command	Server Ownership state. e.g. :SYS:OWN= WaferMapping.frmMap true
:SYS:REG=<apps>	:SYS:REG command	Update which apps are running. The value is a bit field as follows: P200L_Navigator = 0x0001 NP7_Video = 0x0002 NP7_Wafer = 0x0004 NP7_Router = 0x0008

### 7.1.3 netProbe 7 P200L Server TCP/IP Command Set

Command	Description
*IDN?	Get the software ID
:JOY:IL	Turn the illuminator on or off (may require additional hardware)
:JOY:IL?	Get the current state of the illuminator
:JOY:NAVLOCK	Lock out operator navigation functions
:JOY:NAVLOCK?	Get the navigation lock state
:PRB:AL	Set mechanical alignment angle in radians
:PRB:AL?	Get mechanical alignment angle in radians
:PRB:CEN:ALL	Move XYZ to their center positions
:PRB:CEN:VAL	Set the XYZ center position
:PRB:CEN:VAL?	Get the XYZ center position
:PRB:CEN:X	Move the X axis to its center position
:PRB:CEN:Y	Move the Y axis to its center position
:PRB:DN	Lower the platen
:PRB:DNM	Set the auto-z modes for the platen
:PRB:DNM?	Get the auto-z modes for the platen
:PRB:GRID:FILE	Open the grid file
:PRB:GRID:FILE?	Get the grid file name
:PRB:GRID:USE	Enable grid map compensation
:PRB:GRID:USE?	Get grid map compensation enabled
:PRB:HO:ALL	Move XYZ axes to their home positions
:PRB:HO:ALL?	Get homed state of XYZ axes
:PRB:HO:X	Home the X axis
:PRB:HO:X?	Get X axis homed
:PRB:HO:XY	Home the XY axes
:PRB:HO:XY?	Get XY axes homed
:PRB:HO:Y	Home the Y axis
:PRB:HO:Y?	Get Y axis homed
:PRB:HO:Z	Home the Z axis
:PRB:HO:Z?	Get Z axis homed
:PRB:JOG:XY	Jog the XY axes at a percentage of peak speed
:PRB:JOG:Z	Jog the Z axis at a percentage of its peak speed
:PRB:KI	Immediately abort the current command
:PRB:LIM?	Get the state of the XYZ limits
:PRB:MOV:XYA	Move XY axes to a position referenced to user offset
:PRB:MOV:XYNO	Move XY axes to a compensated position with no user offset
:PRB:MOV:XYR	Move XY axes to an uncompensated raw position
:PRB:MOV:XYRE	Move XY axes by a distance
:PRB:MOV:ZA	Move Z axis to a position

:PRB:MOV:ZRE	Move Z axis by a distance
:PRB:OR	Set XY orthogonality
:PRB:OR?	Get XY orthogonality
:PRB:POS:XYZ?	Get XYZ compensated position with user offset
:PRB:POS:XYZNO?	Get XYZ compensated position without user offset
:PRB:POS:XYZR?	Get XYZ uncompensated (raw) position
:PRB:REFXY	Set current position to an XY value
:PRB:REFXY?	Get the user offset
:PRB:STO	Stop any active moves
:PRB:TRA?	Get the XYZ axes travels
:PRB:UDO	Set the Z axis up, down, standby, and overdrive values
:PRB:UDO?	Get the Z axis up, down, standby, and overdrive values
:PRB:UN	Move all axes to their unload positions
:PRB:UN:VAL	Set the unload positions
:PRB:UN:VAL?	Get the unload positions
:PRB:UP	Raise the platen
:SYS:OWN	Take exclusive control of the server
:SYS:OWN?	Get the owned state and the owner
:SYS:REG?	Get running applications
ERR?	Get most recent error
STA?	Get current motion control status
UPDT	Set the update enabled state and mask
UPDT?	Get the update enabled state mask

#### 7.1.4 \*IDN?

**Description:** Return the instrument identification query as a four field, comma delimited string.

**Command Format:** \*IDN?

**Returned Data:** The returned string is in the following format:

<Manufacturer>,<Prober Model>,<Software Model>,<Software Revision>

#### Usage:

Host	P200L_Server
*IDN?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
ID Information	← Micromanipulator,P200L,netProbe P200L Server,1.1.5<CR>
Command is complete	← #064<CR>

#### 7.1.5 :JOY:IL

**Description:** Turn the illuminator on and off.

**Command Format:** :JOY:IL <state>

Where state is either **true** for **ON** or **false** for **OFF**

**Returned Data:** <state> where state is either **True** or **False**

#### Usage:

Host	P200L_Server
:JOY:IL true<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.1.6 :JOY:IL?

**Description:** Get the illuminator on or off state.

**Command Format:** :JOY:IL?

**Returned Data:** <state> where state is either **True for ON** or **False for OFF**

**Usage:**

Host	P200L_Server
:JOY:IL?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Current illuminator state is ON	← True<CR>
Command is complete	← #064<CR>

#### 7.1.7 :JOY:NAVLOCK

**Description:** Set the navigation lock state. Locking navigation prevents the joystick and GUI navigation controls from initiating actions on the prober.

**Command Format:** :JOY:NAVLOCK <flag>

Where flag is either **true** or **false**

**Returned Data:** <state> where state is either **True** or **False**

**Usage:**

Host	P200L_Server
:JOY:NAVLOCK true<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Updated Navlock state is returned	← True<CR>
Command is complete	← #064<CR>

#### 7.1.8 :JOY:NAVLOCK?

**Description:** Get the navigation lock state.

**Command Format:** :JOY:NAVLOCK?

**Returned Data:** <state> where state is either **True** or **False**

**Usage:**

Host	P200L_Server
:JOY:NAVLOCK?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Current Navlock state is returned	← True<CR>
Command is complete	← #064<CR>

#### 7.1.9 :PRB:AL

**Description:** Set the DUT alignment angle referenced to stage XY travel.

**Command Format:** :PRB:AL <enable> <angle>

Where <enable> is either **true** or **false**

<angle> is the alignment angle in Radians in the range  $-\pi/4$  to  $\pi/4$

**Returned Data:** New enabled state and alignment angle.

**Usage:**

Host	P200L_Server
:PRB:AL true 0.0003<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
New alignment settings are returned	← True 0.0003<CR>
Command is complete	← #064<CR>

#### 7.1.10 :PRB:AL? Get mechanical alignment angle in radians

**Description:** Get the DUT alignment angle referenced to stage XY travel.

**Command Format:** :PRB:AL?

**Returned Data:** <enable> <angle>

Where <enable> is either **true** or **false**



<angle> is the alignment angle in Radians in the range  $-\pi/4$  to  $\pi/4$

**Usage:**

Host	P200L_Server
:PRB:AL?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Alignment settings are returned	← True 0.0003<CR>
Command is complete	← #064<CR>

7.1.11 :PRB:CEN:ALL

**Description:** Move XY to their center positions

**Command Format:** :PRB:CEN:ALL

**Returned Data:** <xu> <yu> <zu>

Where <xu> <yu> <zu> are the XYZ uncompensated positions (see section 7.1.46 :PRB:POS:XYZNO?) after the center move is completed. Uncompensated values are returned to indicate the position of the stage relative to the home location.

**Usage:**

Host	P200L_Server
:PRB:CEN:ALL<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Uncompensated XYZ position is returned	← -8696.7939 -3086.5187 0.0000<CR>
Command is complete	← #064<CR>

7.1.12 :PRB:CEN:VAL

**Description:** Set the XY center position

**Command Format:** :PRB:CEN:ALL <x> <y>

Where <x> <y> are the center XY compensated positions (see section 7.1.46 :PRB:POS:XYZNO?). Compensated positions are used because the operator and host are more likely to use locations referenced to a user defined (0,0) than the home position.

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:PRB:CEN:VAL 3000 -3000<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

7.1.13 :PRB:CEN:VAL?

**Description:** Get the XYZ center position

**Command Format:** :PRB:CEN:VAL?

**Returned Data:** <x> <y>

Where <x> <y> are the center XY compensated positions. (see section 7.1.46 :PRB:POS:XYZNO?)

**Usage:**

Host	P200L_Server
:PRB:CEN:VAL?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Compensated XY center positions	← 3000.0000 -3000.0000<CR>
Command is complete	← #064<CR>

7.1.14 :PRB:CEN:X

**Description:** Move the X axis to its center position.

**Command Format:** :PRB:CEN:X

**Returned Data:** <xu>

Where <xu> is the X uncompensated position after the X center move is completed.

**Usage:**

Host	P200L_Server
:PRB:CEN:X<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Uncompensated X position after moving	← -8696.7939<CR>
Command is complete	← #064<CR>

7.1.15 :PRB:CEN:Y

**Description:** Move the Y axis to its center position.

**Command Format:** :PRB:CEN:Y

**Returned Data:** <yu>

Where <yu> is the Y uncompensated position after the Y center move is completed.

**Usage:**

Host	P200L_Server
:PRB:CEN:Y<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Uncompensated Y position after moving	← -3086.5187 <CR>
Command is complete	← #064<CR>

7.1.16 :PRB:DN

**Description:** Lower the platen to the down set point.

**Command Format:** :PRB:DN

**Returned Data:** <z>

Where <z> is the Z position of the platen after the move.

**Usage:**

Host	P200L_Server
<code>:PRB:DN&lt;CR&gt;</code> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Z platen position after the move	← 150.0000<CR>
Command is complete	← #064<CR>

7.1.17 `:PRB:DNM`**Description:** Set the auto-z flags for the platen**Command Format:** `:PRB:DNM <autoraise> <autolower> <standby> <overdrive>`

When enabled,

`<autoraise>` causes the platen to always raise before an XY move`<autolower>` causes the platen to lower after a programmed XY move`<standby>` uses the standby height when raising the platen, rather than a preset up position.`<overdrive>` causes the platen to overdrive down by the overdrive distance when the `:PRB:DN` command is executed.**Returned Data:** Actual auto-z settings**Usage:**

Host	P200L_Server
<code>:PRB:DNM true false true false&lt;CR&gt;</code> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Actual auto-z settings	← True False True False<CR>
Command is complete	← #064<CR>

7.1.18 `:PRB:DNM?` Get the auto-z modes for the platen**Description:** Set the auto-z flags for the platen

**Command Format:** :PRB:DNM?

**Returned Data:** Auto-z settings <autoraise> <autolower> <standby> <overdrive>

When enabled,

<autoraise> causes the platen to always raise before an XY move

<autolower> causes the platen to lower after a programmed XY move

<standby> uses the standby height when raising the platen, rather than a preset up position.

<overdrive> causes the platen to overdrive down by the overdrive distance when the :PRB:DN command is executed.

**Usage:**

Host	P200L_Server
:PRB:DNM? <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Actual auto-z settings	← True False True False<CR>
Command is complete	← #064<CR>

7.1.19 :PRB:GRID:FILE

**Description:** Open the XY grid file. If the file is not valid, the USE flag is set false. The grid file is generated with the P300L\_Test program, and cannot be modified without factory assistance.

**Command Format:** :PRB:GRID:FILE <filename>

**Returned Data:** N/A

The grid file format is:

```
<xref>\t<yref>\t<xerr>\t<yerr>\t<temp>\t<temp>\t<temp3>\t<xindex>\t<yindex>\t<prfound>\t<xoff>\t<yoff>
```

Where \t is the TAB character (0x09)

<xref> <yref> are the calculated XY reference positions for the test location

<xerr> <yerr> are the actual XY positions adjusted by the patrec error

<temp> <temp> <temp3> are unused thermal measurements

<xindex> <yindex> are the XY grid spacing of reference locations

<prfound> indicates if pattern recognition was successful (1) at the location

<xoff> <yoff> are the XY offsets from home for (0,0)

For example,

```
5905.5118  -5905.5118  0.0000      0.2625      3.0  3.0  70.0  0  0
      1  -6003.9370  5998.4240
.
.
.
5905.5118  3937.0079  0.0000      -0.5249      3.0  3.0  70.0  0  0
      1  -6003.9370  5998.4240
```

All position values are in mils (1/1000<sup>th</sup> inch) .

**Usage:**

Host	P200L_Server
:PRB:GRID:FILE stgmap.txt<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

7.1.20 :PRB:GRID:FILE? Get the grid file name

**Description:** Get the name of the active grid file

**Command Format:** :PRB:GRID:FILE?

**Returned Data:** <filename> Grid file name

**Usage:**

Host	P200L_Server
:PRB:GRID:FILE? <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Grid file name	← stgmap.txt<CR>
Command is complete	← #064<CR>

7.1.21 :PRB:GRID:USE

**Description:** Enable grid map compensation. If the grid data is not valid, the USE flag is set false.

**Command Format:** :PRB:GRID:USE <enable>

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:PRB:GRID:USE true <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.1.22 :PRB:GRID:USE?

**Description:** Get grid map compensation enabled.

**Command Format:** :PRB:GRID:USE?

**Returned Data:** <enable> as **True** or **False**

**Usage:**

Host	P200L_Server
:PRB:GRID:USE? <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Grid compensation enabled	← True<CR>
Command is complete	← #064<CR>

#### 7.1.23 :PRB:HO:ALL

**Description:** Move XYZ axes to their home positions

**Command Format:** :PRB:HO:ALL

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:PRB:HO:ALL <CR> →	Carriage return terminated command

Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.1.24 :PRB:HO:ALL?

**Description:** Get homed state of XYZ axes.

**Command Format:** :PRB:HO:ALL?

**Returned Data:** <xhomed> <yhomed> <zhomed> as **True** or **False**

**Usage:**

Host	P200L_Server
:PRB:HO:ALL? <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
All axes are homed	← True True True<CR>
Command is complete	← #064<CR>

#### 7.1.25 :PRB:HO:X Home the X axis

**Description:** Move the X axis to its home position

**Command Format:** :PRB:HO:X

**Returned Data:** **N/A**

**Usage:**

Host	P200L_Server
:PRB:HO:X<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.1.26 :PRB:HO:X?

**Description:** Get homed state of the X axis.

**Command Format:** :PRB:HO:X?



**Returned Data:** <xhomed> as **True** or **False**

**Usage:**

Host	P200L_Server
:PRB:HO:X? <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
X axis is homed	← True<CR>
Command is complete	← #064<CR>

#### 7.1.27 :PRB:HO:XY

**Description:** Move the XY axes to their home positions

**Command Format:** :PRB:HO:XY

**Returned Data:** **N/A**

**Usage:**

Host	P200L_Server
:PRB:HO:XY<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.1.28 :PRB:HO:XY?

**Description:** Get homed state of the XY axes.

**Command Format:** :PRB:HO:XY?

**Returned Data:** <xhomed> <yhomed> as **True** or **False**

**Usage:**

Host	P200L_Server
:PRB:HO:XY? <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>

Data is available	← #192<CR>
XY axes are homed	← True True<CR>
Command is complete	← #064<CR>

#### 7.1.29 :PRB:HO:Y

**Description:** Move the Y axis to its home position

**Command Format:** :PRB:HO:Y

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:PRB:HO:Y<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.1.30 :PRB:HO:Y?

**Description:** Get homed state of the Y axis.

**Command Format:** :PRB:HO:Y?

**Returned Data:** <yhomed> as **True** or **False**

**Usage:**

Host	P200L_Server
:PRB:HO:Y? <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Y axis is homed	← True<CR>
Command is complete	← #064<CR>

#### 7.1.31 :PRB:HO:Z

**Description:** Move the Z axis to its home position

**Command Format:** :PRB:HO:Z

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:PRB:HO:Z<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.1.32 :PRB:HO:Z?

**Description:** Get homed state of the Z axis (platen).

**Command Format:** :PRB:HO:Z?

**Returned Data:** <xhomed> as **True** or **False**

**Usage:**

Host	P200L_Server
:PRB:HO:X? <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Z axis is homed	← True<CR>
Command is complete	← #064<CR>

### 7.1.33 :PRB:JOG:XY

**Description:** Jog the X-Y axes at a percentage of peak speed. The :PRB:STO command must be sent to stop a jog.

**Command Format:** :PRB:JOG:XY <xspeed> <yspeed>

**Returned Data:** <x> <y> <z> positions of axes at start of jog command

**Usage:**

Host	P200L_Server
:PRB:JOG:XY 0.1 0<CR> →	Carriage return terminated command

Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ locations at start of jog	← 198.6470 -136.5299 127.6790<CR>
Command is complete	← #064<CR>

#### 7.1.34 :PRB:JOG:Z

**Description:** Jog the Z axis at a percentage of its peak speed. The :PRB:STO command must be sent to stop a jog.

**Command Format:** :PRB:JOG:Z <zspeed>

**Returned Data:** <z> position of platen at start of jog command

**Usage:**

Host	P200L_Server
:PRB:JOG:Z 0.1<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Z location at start of jog	← 127.6790<CR>
Command is complete	← #064<CR>

#### 7.1.35 :PRB:KI

**Description:** Immediately abort the current command

**Command Format:** :PRB:KI

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:PRB:KI<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

7.1.36 :PRB:LIM? Get the state of the XYZ limits

**Description:** Get the current limit states for all axes

**Command Format:** :PRB:LIM?

**Returned Data:** <bitfield> binary field of activated limits as follows:

Bit(value)	7 (128)	6 (64)	5 (32)	4 (16)	3 (8)	2 (4)	1 (2)	0 (1)
Limit	N/A	N/A	-Z	+Z	-Y	+Y	-X	+X

For

example, when the XYZ home limits are active the returned value is 1+8+32 = 41.

**Usage:**

Host	P200L_Server
:PRB:LIM?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
+X, -Y, and -Z limits active	← 41<CR>
Command is complete	← #064<CR>

7.1.37 :PRB:MOV:XYA

**Description:** Move the XY axes to a position referenced to user (0,0)

**Command Format:** :PRB:MOV:XYA <x> <y>

**Returned Data:** <x> <y> <z> positions of axes at end of move

**Usage:**

Host	P200L_Server
:PRB:MOV:XYA 123 321<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ positions after move completed	← 123.0000 321.0000 128.3133<CR>
Command is complete	← #064<CR>

### 7.1.38 :PRB:MOV:XYNO

**Description:** Move XY axes to a compensated position with no user offset

**Command Format:** :PRB:MOV:XYNO <xno> <yno>

**Returned Data:** <x> <y> <z> positions of axes at end of move referenced to user (0,0).

**Usage:**

Host	P200L_Server
:PRB:MOV:XYNO 123 321<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ positions after move completed	← -531.2221 498.9314 128.3133<CR>
Command is complete	← #064<CR>

### 7.1.39 :PRB:MOV:XYR

**Description:** Move XY axes to an uncompensated raw position

**Command Format:** :PRB:MOV:XYR <xraw> <yraw>

**Returned Data:** <x> <y> <z> positions of axes at end of move referenced to user (0,0).

**Usage:**

Host	P200L_Server
:PRB:MOV:XYR -4000 4000<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ positions after move completed	← 0.0000 0.0000 128.3133<CR>
Command is complete	← #064<CR>

### 7.1.40 :PRB:MOV:XYRE

**Description:** Move the XY axes by a distance.

**Command Format:** :PRB:MOV:XYRE <xrel> <yrel>

**Returned Data:** <x> <y> <z> positions of axes at end of move referenced to user (0,0).

**Usage:**

Host	P200L_Server
:PRB:MOV:XYRE 100 0<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ positions at end of move	← 100.0000 0.0000 128.3133<CR>
Command is complete	← #064<CR>

#### 7.1.41 :PRB:MOV:ZA

**Description:** Move Z axis to a position.

**Command Format:** :PRB:MOV:ZA <z>

**Returned Data:** <z> position of Z axis at end of move.

**Usage:**

Host	P200L_Server
:PRB:MOV:ZA 127<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Z position at end of move	← 127.0000<CR>
Command is complete	← #064<CR>

#### 7.1.42 :PRB:MOV:ZRE

**Description:** Move Z axis by a distance.

**Command Format:** :PRB:MOV:ZRE <zrel>

**Returned Data:** <zrel> position of Z axis at end of move.

**Usage:**

Host	P200L_Server
------	--------------

<code>:PRB:MOV:ZRE 5&lt;CR&gt;</code> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Z position at end of move	← 132.0000<CR>
Command is complete	← #064<CR>

#### 7.1.43 :PRB:OR

**Description:** Set XY orthogonality.

**Command Format:** `:PRB:OR <enable> <angle>`

Where <enable> is true or false and <angle> is in the range 1.57079633 +/- .25

**Returned Data:** Updated orthogonality settings.

**Usage:**

Host	P200L_Server
<code>:PRB:OR true 1.571032&lt;CR&gt;</code> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
New orthogonality settings	← True 1.57103200<CR>
Command is complete	← #064<CR>

#### 7.1.44 :PRB:OR?

**Description:** Get XY orthogonality.

**Command Format:** `:PRB:OR?`

**Returned Data:** <enable> <angle>

Where <enable> is true or false and <angle> is in the range 1.57079633 +/- .25

**Usage:**

Host	P200L_Server
<code>:PRB:OR?&lt;CR&gt;</code> →	Carriage return terminated command



Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
New orthogonality settings	← True 1.57103200<CR>
Command is complete	← #064<CR>

#### 7.1.45 :PRB:POS:XYZ?

**Description:** Get XYZ compensated position referenced to user (0,0)

**Command Format:** :PRB:POS:XYZ?

**Returned Data:** <x> <y> <z> positions reference to user (0,0)

**Usage:**

Host	P200L_Server
:PRB:POS:XYZ?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ positions	← 123.0000 321.0000 128.3133<CR>
Command is complete	← #064<CR>

#### 7.1.46 :PRB:POS:XYZNO?

**Description:** Get XYZ compensated position referenced to home position

**Command Format:** :PRB:POS:XYZNO?

**Returned Data:** <xno> <yno> <zno> positions referenced to home position

**Usage:**

Host	P200L_Server
:PRB:POS:XYZNO?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ positions	← -3877.0000 4321.0000 128.3133<CR>

Command is complete	← #064<CR>
---------------------	------------

#### 7.1.47 :PRB:POS:XYZR?

**Description:** Get XYZ uncompensated (raw) position.

**Command Format:** :PRB:POS:XYZR?

**Returned Data:** <xraw> <yraw> <zraw> uncompensated (raw) positions referenced to home position.

#### Usage:

Host	P200L_Server
:PRB:POS:XYZR?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ raw positions	← -3874.1255 4319.7534 128.3133<CR>
Command is complete	← #064<CR>

#### 7.1.48 :PRB:REFXY

**Description:** Set current position to an XY value

**Command Format:** :PRB:REFXY <x> <y>

**Returned Data:** New X-Y position.

#### Usage:

Host	P200L_Server
:PRB:REFXY 0 0<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XY position	← 0.0000 0.0000<CR>
Command is complete	← #064<CR>

#### 7.1.49 :PRB:REFXY?

**Description:** Get the user offset

**Command Format:** :PRB:REFXY?

**Returned Data:** X-Y offset of user (0,0) from home position.

**Usage:**

Host	P200L_Server
:PRB:REFXY?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Raw XY offset	← -4000.0000 4000.0000<CR>
Command is complete	← #064<CR>

#### 7.1.50 :PRB:STO

**Description:** Stop any active moves. This command must be sent to stop JOG commands.

**Command Format:** :PRB:STO

**Returned Data:** XYZ position after move is stopped.

**Usage:**

Host	P200L_Server
:PRB:STO<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ position after move is stopped.	← -4230.1233 4000.0000 128.3423<CR>
Command is complete	← #064<CR>

#### 7.1.51 :PRB:TRA?

**Description:** Get the XYZ axes travels

**Command Format:** :PRB:TRA?

**Returned Data:** XYZ available travel.

**Usage:**

Host	P200L_Server
:PRB:TRA?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
XYZ travels	← 8000.0000 8000.0000 700.0000<CR>
Command is complete	← #064<CR>

## 7.1.52 :PRB:UDO

**Description:** Set the Z axis up, down, standby, and overdrive values

**Command Format:** :PRB:UDO <up> <down> <standby> <overdrive>

Where <up> is the platen destination when a :PRB:UP command is sent.

<down> is the down destination when a :PRB:DN command is sent.

<standby> is the height above the down position to move when the use standby height mode is enabled.

<overdrive> is the distance past the down position to move the platen when the overdrive mode is enabled.

**Returned Data:** New settings.

**Usage:**

Host	P200L_Server
:PRB:UDO 100 150 50 1<CR> →	Carriage return terminated command
Server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Updated settings	← 100.0000 150.0000 50.0000 1.0000<CR>
Command is complete	← #064<CR>

## 7.1.53 :PRB:UDO?

**Description:** Get the Z axis up, down, standby, and overdrive values

**Command Format:** :PRB:UDO?

**Returned Data:** <up> <down> <standby> <overdrive>

Where <up> is the platen destination when a :PRB:UP command is sent.  
 <down> is the down destination when a :PRB:DN command is sent.  
 <standby> is the height above the down position to move when the use standby height mode is enabled.  
 <overdrive> is the distance past the down position to move the platen when the overdrive mode is enabled.

**Usage:**

Host	P200L_Server
: PRB : UDO ? <CR> →	Carriage return terminated command
Server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Current settings	← 100.0000 150.0000 50.0000 1.0000<CR>
Command is complete	← #064<CR>

7.1.54 :PRB:UN

**Description:** Move all axes to their unload positions.

**Command Format:** : PRB : UN

**Returned Data:** <x> <y> <z> positions after unload move referenced to user (0,0)

**Usage:**

Host	P200L_Server
: PRB : UN <CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Position after move is complete	← 0.0000 3700.0000 0.0000<CR>
Command is complete	← #064<CR>

7.1.55 :PRB:UN:VAL

**Description:** Set the XYZ unload position referenced to home location

**Command Format:** : PRB : UN : VAL <xunl> <yunl> <zunl>

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:PRB:UN:VAL -4000 7700 10<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.1.56 :PRB:UN:VAL?

**Description:** Get the unload positions referenced to home

**Command Format:** :PRB:UN:VAL?

**Returned Data:** <xunl> <yunl> <zunl> positions unload positions referenced to home

**Usage:**

Host	P200L_Server
:PRB:UN:VAL?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Position after move is complete	← -4000.0000 7700.0000 10.0000<CR>
Command is complete	← #064<CR>

#### 7.1.57 :PRB:UP

**Description:** Raise the platen to either the up position or by the standby height.

**Command Format:** :PRB:UP

**Returned Data:** <z>

Where <z> is the Z position of the platen after the move.

**Usage:**

Host	P200L_Server
:PRB:UP<CR> →	Carriage return terminated command

Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Z platen position after the move	← 100.0000<CR>
Command is complete	← #064<CR>

### 7.1.58 :SYS:OWN

**Description:** Take exclusive control of the server

**Command Format:** :SYS:OWN <ownerid> <owned>

Where <ownerid> is a text string descriptor

<owned> is true or false indicating ownership

**Returned Data:** Name of owner and if the server is owned.

**Usage:**

Host	P200L_Server
:SYS:OWN myapp true<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Name of owner and owned state	← myapp True<CR>
Command is complete	← #064<CR>

### 7.1.59 :SYS:OWN? Get the owned state and the owner

**Description:** Get the owned state of the server.

**Command Format:** :SYS:OWN?

**Returned Data:** <ownerid> <owned>

Where <ownerid> is name of owner. If the server is not owned, the owner name is <none>.

<owned> is true or false

**Usage:**

Host	P200L_Server
:SYS:OWN?<CR> →	Carriage return terminated command

Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Name of owner and owned state	← myapp True<CR>
Command is complete	← #064<CR>

#### 7.1.60 :SYS:REG?

**Description:** Get running applications.

**Command Format:** :SYS:REG?

**Returned Data:** <bitfield> where the bit values are as follows:

Bit(value)	7 (128)	6 (64)	5 (32)	4 (16)	3 (8)	2 (4)	1 (2)	0 (1)
Limit	N/A	N/A	N/A	N/A	Router	Wafer	Video	Navigator

**Usage:**

Host	P200L_Server
:SYS:REG?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Navigator and Video are running	← 3<CR>
Command is complete	← #064<CR>

#### 7.1.61 ERR?

**Description:** Get most recent error

**Command Format:** ERR?

**Returned Data:** <errorcode>:<description>

**Usage:**

Host	P200L_Server
ERR?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>



Data is available	← #192<CR>
Navigator and Video are running	← 0:No Errors<CR>
Command is complete	← #064<CR>

### 7.1.62 STA? Get current motion control status

**Description:** Get current motion status

**Command Format:** STA?

**Returned Data:** <statuscode>:<description>

**Usage:**

Host	P200L_Server
STA?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
No active motion	← 0:Idle<CR>
Command is complete	← #064<CR>

### 7.1.63 UPDT

**Description:** Set the update enabled state and mask

**Command Format:** UPDT <bitfield>

Where <bitfield> is as follows:

Bit	Decimal (Hex) Value	Description
0	1 (0x0001)	Enable Updates
3	8 (0x0008)	Joystick Navlock (:JOY:NAVLOCK=true   false)
4	16 (0x0010)	Joystick Jogging (:JOY:JOG=true   false true   false true   false)
8	256 (0x0100)	Device position before, during, and after joystick jog (:PRB:POS=<x> <y> <z>)

**Returned Data:** <errorcode>:<description>

**Usage:**

Host	P200L_Server
UPDT 9<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.1.64 UPDT?

**Description:** Get the update enabled state mask

**Command Format:** UPDT?

**Returned Data:** <bitfield> as described under the UPDT command

#### Usage:

Host	P200L_Server
UPDT?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Updates enabled for joystick navlock	← 9<CR>
Command is complete	← #064<CR>

## 7.2 Navigator TCP/IP Commands

Command	Description
:SUB:CURR?	Get the current sub-site location information
:SUB:FILE	Open a sub-site file
:SUB:FILE?	Get the current sub-site file name
:SUB:FIRST?	Move to the first sub-site step location
:SUB:GOTO	Move to a specific sub-site step
:SUB:NEXT?	Move to the next sub-site step location
:SUB:PREV?	Move to the previous sub-site step location
:SUB:RESET	Reset the sub-site step counter
:SYS:OWN	Take sole control of the interface
:SYS:OWN?	Get current owner and owned state

### 7.2.1 :SUB:CURR?

**Description:** Get the current sub-site location information. If the Sub-Die program has not started or a step is not active, an error status is returned.

**Command Format:** :SUB:CURR?

**Returned Data:** <step> <x> <y> <enabled> <units>

Where <step> is the active step number. If no steps are selected an error is returned.  
 <x> and <y> are the positions associated with the step in mm.  
 <enabled> indicates if the step is included when running through the list.  
 <units> is the units of <x> and <y>. Values are “mm”, “ $\mu$ m”, and “mils”;

### Usage:

Host	P200L_Server
:SUB:CURR?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Step information	← 1 0.0000 0.0000 True mm<CR>
Command is complete	← #064<CR>

### 7.2.2 :SUB:FILE Open a sub-site file

**Description:** Open a sub-site file. If the file is not found an error status is returned.

**Command Format:** :SUB:FILE <filename>

Where <filename> is the file to open. The file format is either Excel CSV or XML.

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:SUB:CURR mysubdie.csv<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.2.3 :SUB:FILE?

**Description:** Get the current sub-site location information

**Command Format:** :SUB:FILE?

**Returned Data:** <filename> of the active Sub-Site List

**Usage:**

Host	P200L_Server
:SUB:FILE?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Active file name	← C:\SubSites\test1.csv<CR>
Command is complete	← #064<CR>

### 7.2.4 :SUB:FIRST?

**Description:** Move to the first sub-site step location

**Command Format:** :SUB:FIRST?

**Returned Data:** <step> <x> <y> <enabled> <units>

Where <step> is the active step number. If no steps are selected an error is returned.  
 <x> and <y> are the positions associated with the step in mm.  
 <enabled> indicates if the step is included when running through the list.  
 <units> is the units of <x> and <y>. Values are “mm”, “µm”, and “mils”;

**Usage:**

Host	P200L_Server
<code>:SUB:FIRST?&lt;CR&gt;</code> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Step information	← 1 0.0000 0.0000 True mm<CR>
Command is complete	← #064<CR>

7.2.5 `:SUB:GOTO`**Description:** Move to a specific sub-site step**Command Format:** `:SUB:GOTO <step>`

Where <step> is the step number in the list to move to. If the step is out of range an error is returned.

**Returned Data:** `<step> <x> <y> <enabled> <units>`

Where `<step>` is the active step number. If no steps are selected an error is returned.  
`<x>` and `<y>` are the positions associated with the step in mm.  
`<enabled>` indicates if the step is included when running through the list.  
`<units>` is the units of `<x>` and `<y>`. Values are “mm”, “μm”, and “mils”;

**Usage:**

Host	P200L_Server
<code>:SUB:GOTO 1&lt;CR&gt;</code> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Step information	← 1 0.0000 0.0000 True mm<CR>
Command is complete	← #064<CR>

7.2.6 `:SUB:NEXT?`**Description:** Move to the next sub-site step location**Command Format:** `:SUB:NEXT?`

**Returned Data:** <step> <x> <y> <enabled> <units>

Where <step> is the active step number. If no steps are selected an error is returned.

If the current step is the last step in the list, a -1 is returned.

<x> and <y> are the positions associated with the step in mm.

<enabled> indicates if the step is included when running through the list.

<units> is the units of <x> and <y>. Values are “mm”, “μm”, and “mils”;

**Usage:**

Host	P200L_Server
:SUB:NEXT?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Step information	← 2 0.0000 0.1000 True mm<CR>
Command is complete	← #064<CR>

7.2.7 :SUB:PREV?

**Description:** Move to the previous sub-site step location

**Command Format:** :SUB:PREV?

**Returned Data:** <step> <x> <y> <enabled> <units>

Where <step> is the active step number. If no steps are selected an error is returned.

<x> and <y> are the positions associated with the step in mm.

<enabled> indicates if the step is included when running through the list.

<units> is the units of <x> and <y>. Values are “mm”, “μm”, and “mils”;

**Usage:**

Host	P200L_Server
:SUB:PREV?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Step information	← 1 0.0000 0.0000 True mm<CR>
Command is complete	← #064<CR>

### 7.2.8 :SUB:RESET

**Description:** Reset the sub-site step counter

**Command Format:** :SUB:RESET

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:SUB:RESET<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.2.9 :SYS:OWN

**Description:** Take exclusive control of the Sub-Site server

**Command Format:** :SYS:OWN <ownerid> <owned>

Where <ownerid> is a text string descriptor

<owned> is true or false indicating ownership

**Returned Data:** Name of owner and if the server is owned.

**Usage:**

Host	P200L_Server
:SYS:OWN myapp true<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Name of owner and owned state	← myapp True<CR>
Command is complete	← #064<CR>

### 7.2.10 :SYS:OWN?

**Description:** Get the owned state of the Sub-Site server.

**Command Format:** :SYS:OWN?

**Returned Data:** <ownerid> <owned>

Where <ownerid> is name of owner. If the server is not owned, the owner name is <none>.

<owned> is true or false

**Usage:**

Host	P200L_Server
:SYS:OWN?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Name of owner and owned state	← myapp True<CR>
Command is complete	← #064<CR>



### 7.3 Wafer Map TCP/IP Commands

Command	Description
:SYS:OWN	Take ownership of Wafer Map
:SYS:OWN?	Get ownership status of Wafer map
:WFR:AUTOIL	Enable/Disable turning illuminator on during, and off after moves
:WFR:AUTOIL?	Get the illuminator auto state
:WFR:BIN	Set the Bincodes of a die
:WFR:BIN:LIST?	Get the list of available Bincodes. e.g. TGX0I for "Test", "Good", "Fail", "Null", and "Invalid". The "I" bincodes are used to indicate that the die at the location is unusable or absent.
:WFR:BIN:VAL	Get the bincodes for a specific die.
:WFR:CURR?	Get the Column,Row of the die under test
:WFR:DIESIZE	Set the die size in mm.
:WFR:DIESIZE?	Get the die size in mm.
:WFR:FILE	Open a wafer map file
:WFR:FILE?	Get the name of the active wafer map file
:WFR:FIRST:VAL?	Get the Column, Row of the first die in the test.
:WFR:FIRST?	Move to the first test die and return its location.
:WFR:MAP:COUNT?	Get the number of testable die on the wafer
:WFR:MAP:CR?	Get the number of rows and columns in the map.
:WFR:MAP:CSV	Send comma separated text to use as the active map.
:WFR:MAP:CSV?	Receive comma separated text defining the active map.
:WFR:MAP:XML	Send XML formatted text to use as the active map.
:WFR:MAP:XML?	Receive XML formatted text defining the active map.
:WFR:MAP:XY	Get the home referenced XY location associated with the CR location.
:WFR:MOV:CR	Move to a Column, Row
:WFR:NEXT?	Move to next testable die and return its location
:WFR:PATREC	Enable/Disable pattern recognition.
:WFR:PATREC?	Get Enabled/Disabled state of pattern recognition.
:WFR:POS:CR?	Get current Column, Row position
:WFR:REFCR	Set the Column, Row position
:WFR:REFCR?	Get the X,Y offset of the zeroth Column, Row. This offset is different than the user defined X,Y offset available in Navigator
:WFR:RESET	Reset the bincodes of active die to "Test" and clear the map test cursor
:WFR:SUBSITES	Enable/Disable Navigator Sub-Site tests at each die
:WFR:SUBSITES?	Get Enabled/Disabled state of Navigator Sub-Site tests

### 7.3.1 :SYS:OWN

**Description:** Take exclusive control of the wafer map server

**Command Format:** :SYS:OWN <ownerid> <owned>

Where <ownerid> is a text string descriptor

<owned> is true or false indicating ownership

**Returned Data:** Name of owner and if the server is owned.

**Usage:**

Host	P200L_Server
:SYS:OWN myapp true<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Name of owner and owned state	← myapp True<CR>
Command is complete	← #064<CR>

### 7.3.2 :SYS:OWN?

**Description:** Get the owned state of the wafer map server.

**Command Format:** :SYS:OWN?

**Returned Data:** <ownerid> <owned>

Where <ownerid> is name of owner. If the server is not owned, the owner name is <none>.

<owned> is true or false

**Usage:**

Host	P200L_Server
:SYS:OWN?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Name of owner and owned state	← myapp True<CR>
Command is complete	← #064<CR>

### 7.3.3 :WFR:AUTOIL

**Description:** Enable/Disable automatically turning on the illuminator before die moves, and turning the illuminator off after the move is completed.

**Command Format:** :WFR:AUTOIL <state>

**Returned Data:** N/A

#### Usage:

Host	P200L_Server
:WFR:AUTOIL true →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.3.4 :WFR:AUTOIL?

**Description:** Get Enabled/Disabled state of auto illumination.

**Command Format:** :WFR:AUTOIL?

**Returned Data:** <enabled> state

#### Usage:

Host	P200L_Server
:WFR:AUTOIL?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned state	True<CR>
Command is complete	← #064<CR>

### 7.3.5 :WFR:BIN

**Description:** Set the bin code for a die on the wafer map

**Command Format:** :WFR:BIN <col> <row> <bincode>

Where <col> is the column of the die

<row> is the row of the die

<bincode> is one of the following:

Ø=Untestable G=Passed X=Failed I=Ignore T=Untested

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:WFR:BIN 3 4 G<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.3.6 :WFR:BIN:LIST?

**Description:** Get the available bin codes for die on the wafer map

**Command Format:** :WFR:BIN:LIST?

**Returned Data:** <list> as a 5 character string with characters in the following order:

String Index	Default Value	Description
0	T	Untested
1	G	Passed Test
2	X	Failed Test
3	Ø	Untestable
4	I	Ignore die during test

**Usage:**

Host	P200L_Server
:WFR:BIN:LIST?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned bin codes	TGXØI<CR>
Command is complete	← #064<CR>

### 7.3.7 :WFR:BIN:VAL

**Description:** Get the bincode for specific die on the wafer map

**Command Format:** :WFR:BIN:VAL <col> <row>

Where <row>,<col> are the die location on the map

**Returned Data:** <row> <col> <bincode> for the die

where <row>,<col> is the requested the die location on the map

<bincode> is 0=Untestable G=Passed X=Failed I=Ignore T=Untested

**Usage:**

Host	P200L_Server
:WFR:BIN:VAL 3 4 d<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned bincode	3 4 G<CR>
Command is complete	← #064<CR>

### 7.3.8 :WFR:CURRE?

**Description:** Get the current die under test. If the test has not started or the wafer map

**Command Format:** :WFR:CURRE?

**Returned Data:** <col> <row> die position

**Usage:**

Host	P200L_Server
:WFR:CURRE?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned location	2 0<CR>
Command is complete	← #064<CR>

### 7.3.9 :WFR:DIESIZE

**Description:** Set the die size in mm

**Command Format:** :WFR:DIESIZE <xdie> <ydie>

Where <xdie> is the horizontal die spacing

<ydie> is the vertical die spacing

**Returned Data:** <xdie> <ydie>

**Usage:**

Host	P200L_Server
:WFR:DIESIZE 10 10<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
New diesize	10.0000 10.0000<CR>
Command is complete	← #064<CR>

7.3.10 :WFR:DIESIZE?

**Description:** Get the current die size in mm

**Command Format:** :WFR:DIESIZE?

**Returned Data:** <xdie> <ydie>

Where <xdie> is the horizontal die spacing

<ydie> is the vertical die spacing

**Usage:**

Host	P200L_Server
:WFR:DIESIZE?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned die size	10.0000 10.0000<CR>
Command is complete	← #064<CR>

7.3.11 :WFR:FILE

**Description:** Open a wafer map file

**Command Format:** :WFR:FILE <filename>

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:WFR:FILE nrml.csv<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

7.3.12 :WFR:FILE?

**Description:** Get the name of the active wafer map file

**Command Format:** :WFR:FILE?

**Returned Data:** <filename>

**Usage:**

Host	P200L_Server
:WFR:FILE?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned file name	c:\mapfiles\nrml.csv<CR>
Command is complete	← #064<CR>

7.3.13 :WFR:FIRST:VAL?

**Description:** Get the Column, Row of the first die in the test. The first die is determined by the start corner and test pattern.

**Command Format:** :WFR:FIRST:VAL?

**Returned Data:** <col> <row> of first die in the test

**Usage:**

Host	P200L_Server
:WFR:FIRST:VAL?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>

Returned location	6 0<CR>
Command is complete	← #064<CR>

#### 7.3.14 :WFR:FIRST?

**Description:** Move to the first test die and return the stage location after the move.

**Command Format:** :WFR:FIRST?

**Returned Data:** <col> <row> of stage location after the move

#### Usage:

Host	P200L_Server
:WFR:FIRST?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned position	6 0<CR>
Command is complete	← #064<CR>

#### 7.3.15 :WFR:MAP:COUNT?

**Description:** Get the number of testable die on the wafer. This is a useful value for some programs which test a wafer by moving a specific number of steps.

**Command Format:** :WFR:MAP:COUNT?

**Returned Data:** <count> of testable die

#### Usage:

Host	P200L_Server
:WFR:MAP:COUNT?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned testable die count	293<CR>
Command is complete	← #064<CR>



### 7.3.16 :WFR:MAP:CR?

**Description:** Get the number of rows and columns in the map.

**Command Format:** :WFR:MAP:CR?

**Returned Data:** <columns> <rows> on wafer

**Usage:**

Host	P200L_Server
:WFR:MAP:CR?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned columns and rows	20 20<CR>
Command is complete	← #064<CR>

### 7.3.17 :WFR:MAP:CSV

**Description:** Send comma separated text to use as the active map.

**Command Format:** :WFR:MAP:CSV <formatted\_text>

Where <formatted\_text> is a single string in the same format as the CSV file. The character '~' is used in place of CR or CRLF.

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:WFR:MAP:CSV SubstrateType,Wafer~Units,mm~WaferSize,200.0000~Rows,20~ Columns,20~DeviceSizeX,10.0000~DeviceSizeY,10.0000~ ExclusionBand,0.0000~NullBin,0~Bin,G~Bin,X~Bin,I~ Orientation,0~OriginLocation,0~Spline,0~ Row 0,IIIIIIIGTTTTTTTTIIIIIII~ Row 1,IIIIITTTTTTTTTTTTTIIIIII~ Row 2,IIITTTTTTTTTTTTTTTTTIIII~ Row 3,IITTTTTTTTTTTTTTTTTTIIII~ Row 4,ITTTTTTTTTTTTTTTTTTTII~ Row 5,ITTTTTTTTTTTTTTTTTTTII~ Row 6,TTTTTTTTTTTTTTTTTTTTII~ Row 7,TTTTTTTTTTTTTTTTTTTTII~ Row 8,TTTTTTTTTTTTTTTTTTTTII~	Carriage return terminated command

<pre> Row 9,TTTTTTTTTTTTTTTTTTTTTI~ Row 10,TTTTTTTTTTTTTTTTTTTTTI~ Row 11,TTTTTTTTTTTTTTTTTTTTTI~ Row 12,TTTTTTTTTTTTTTTTTTTTTI~ Row 13,ITTTTTTTTTTTTTTTTTTII~ Row 14,ITTTTTTTTTTTTTTTTTTII~ Row 15,IITTTTTTTTTTTTTTTTTIII~ Row 16,IIITTTTTTTTTTTTTTTTTIII~ Row 17,IIIITTTTTTTTTTTTIIIII~ Row 18,IIIIITTTTTTTTTIIIIIII~ Row 19,IIIIIIIIIIIIIIIIIIII&lt;CR&gt; → </pre>	
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.3.18 :WFR:MAP:CSV?

**Description:** Receive active map as comma separated text.

**Command Format:** :WFR:MAP:CSV?

**Returned Data:** <formatted\_text>

Where <formatted\_text> is a single string in the same format as the CSV file. The character '~' is used in place of CR or CRLF.

#### Usage:

Host	P200L_Server
:WFR:MAP:CSV?<CR>→	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned CSV text	<pre> SubstrateType,Wafer~Units,mm~WaferSize,200.0000~Rows,20~ Columns,20~DeviceSizeX,10.0000~DeviceSizeY,10.0000~ ExclusionBand,0.0000~NullBin,0~Bin,G~Bin,X~Bin,I~ Orientation,0~OriginLocation,0~Spline,0~ Row 0,IIIIIGTTTTTTTTIIIIII~ Row 1,IIIIITTTTTTTTTIIIIII~ Row 2,IIITTTTTTTTTTTTTIIII~ Row 3,IITTTTTTTTTTTTTTTTTIII~ Row 4,ITTTTTTTTTTTTTTTTTTII~ Row 5,ITTTTTTTTTTTTTTTTTTII~ Row 6,TTTTTTTTTTTTTTTTTTTTI~ Row 7,TTTTTTTTTTTTTTTTTTTTI~ Row 8,TTTTTTTTTTTTTTTTTTTTI~ Row 9,TTTTTTTTTTTTTTTTTTTTI~ Row 10,TTTTTTTTTTTTTTTTTTTTI~ Row 11,TTTTTTTTTTTTTTTTTTTTI~ </pre>

	Row 12, TTTTTTTTTTTTTTTTTTII~ Row 13, ITTTTTTTTTTTTTTTTTII~ Row 14, ITTTTTTTTTTTTTTTTTII~ Row 15, IITTTTTTTTTTTTTTTTTIII~ Row 16, IIIITTTTTTTTTTTTTIIIII~ Row 17, IIIITTTTTTTTTTTTTIIIII~ Row 18, IIIIIITTTTTTTTTIIIIIII~ Row 19, IIIIIITTTTTTTTTIIIIIII<CR>
Command is complete	← #064<CR>

### 7.3.19 :WFR:MAP:XML

**Description:** Send XML formatted text to use as the active map.

**Command Format:** :WFR:MAP:XML <formatted\_text>

Where <formatted\_text> is a single string in the same format as the XML file. The character '~' is used in place of CR or CRLF.

**Returned Data:** N/A

### Usage:

Host	P200L_Server
:WFR:MAP:CSV <Maps>~ <Map SubstrateType="Wafer">~ <Device Rows="20" Columns="20" NullBin="0" WaferSize="200.0000" DeviceSizeX="10000.0000" DeviceSizeY="10000.0000" Units="mm" Orientation="0" OriginLocation="0">~ <ReferenceDevice ReferenceDeviceX="10" ReferenceDeviceY="10" />~ <Data>~ <Row><! [CDATA[ IIIIIIGTTTTTTIIIIIII ]></Row>~ <Row><! [CDATA[ IIIITTTTTTTTTTTIIIIII ]></Row>~ <Row><! [CDATA[ IITTTTTTTTTTTTTTTIIII ]></Row>~ <Row><! [CDATA[ IITTTTTTTTTTTTTTTIIII ]></Row>~ <Row><! [CDATA[ ITTTTTTTTTTTTTTTTTTII ]></Row>~ <Row><! [CDATA[ ITTTTTTTTTTTTTTTTTTII ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ TTTTTTTTTTTTTTTTTTTI ]></Row>~ <Row><! [CDATA[ ITTTTTTTTTTTTTTTTTTII ]></Row>~ <Row><! [CDATA[ ITTTTTTTTTTTTTTTTTTII ]></Row>~ <Row><! [CDATA[ IITTTTTTTTTTTTTTTTTIII ]></Row>~ <Row><! [CDATA[ IITTTTTTTTTTTTTTTTTIII ]></Row>~ <Row><! [CDATA[ IIIITTTTTTTTTTTTTIIIII ]></Row>~ <Row><! [CDATA[ IIIITTTTTTTTTTTTTIIIII ]></Row>~ <Row><! [CDATA[ IIIIIITTTTTTTTTIIIIIII ]></Row>~ <Row><! [CDATA[ IIIIIITTTTTTTTTIIIIIII ]></Row>~	Carriage return terminated command

	<pre> &lt;/Data&gt;~ &lt;/Device&gt;~ &lt;/Map&gt;~ &lt;/Maps&gt; &lt;CR&gt; → </pre>
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.3.20 :WFR:MAP:XML?

**Description:** Receive XML formatted text defining the active map.

**Command Format:** :WFR:MAP:XML?

**Returned Data:** <formatted\_text>

Where <formatted\_text> is a single string in the same format as the XML file. The character '~' is used in place of CR or CRLF.

### Usage:

Host	P200L_Server
:WFR:MAP:CSV? →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned XML text	<pre> &lt;Maps&gt;~ &lt;Map SubstrateType="Wafer"&gt;~ &lt;Device Rows="20" Columns="20" NullBin="0" WaferSize="200.0000" DeviceSizeX="10000.0000" DeviceSizeY="10000.0000" Units="mm" Orientation="0" OriginLocation="0"&gt;~ &lt;ReferenceDevice ReferenceDeviceX="10" ReferenceDeviceY="10" /&gt;~ &lt;Data&gt;~ &lt;Row&gt;&lt;! [CDATA [ I I I I I I G T T T T T T T I I I I I I I I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ I I I I T T T T T T T T T T T T I I I I I I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ I I I T T T T T T T T T T T T T T I I I I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ I I T T T T T T T T T T T T T T T T I I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ I T T T T T T T T T T T T T T T T T T I I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ I T T T T T T T T T T T T T T T T T T I I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ T T T T T T T T T T T T T T T T T T T I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ T T T T T T T T T T T T T T T T T T T I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ T T T T T T T T T T T T T T T T T T T I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ T T T T T T T T T T T T T T T T T T T I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ T T T T T T T T T T T T T T T T T T T I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ T T T T T T T T T T T T T T T T T T T I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ T T T T T T T T T T T T T T T T T T T I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ I T T T T T T T T T T T T T T T T T T I I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ I T T T T T T T T T T T T T T T T T T I I ] ] &gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;! [CDATA [ I I T T T T T T T T T T T T T T T T I I I I ] ] &gt;&lt;/Row&gt;~ </pre>

	<pre>&lt;Row&gt;&lt;![CDATA[IIIIITTTTTTTTTTIIII]]&gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;![CDATA[IIIIITTTTTTTTTTIIII]]&gt;&lt;/Row&gt;~ &lt;Row&gt;&lt;![CDATA[IIIIIIIIIIIIIIIIII]]&gt;&lt;/Row&gt;~ &lt;/Data&gt;~ &lt;/Device&gt;~ &lt;/Map&gt;~ &lt;/Maps&gt;&lt;CR&gt;</pre>
Command is complete	← #064<CR>

### 7.3.21 :WFR:MAP:XY

**Description:** Get the home referenced XY location associated with the CR location.

**Command Format:** :WFR:MAP:XY <col> <row>

**Returned Data:** <xpos> <ypos> location corresponding to the <col> <row> position

Where <xpos> <ypos> are referenced to their home locations. The reason for using the home reference rather than the user (0,0) is that when the map is run with the Sub-Site list, the user (0,0) is reset at each die.

#### Usage:

Host	P200L_Server
:WFR:MAP:XY 6 0<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned position	-3214.0144 3149.6061<CR>
Command is complete	← #064<CR>

### 7.3.22 :WFR:MOV:CR

**Description:** Move to a Column, Row.

**Command Format:** :WFR:MOV:CR <col> <row>

Where <col> <row> is the destination of the move

**Returned Data:** <col> <row> of stage after the move

#### Usage:

Host	P200L_Server
:WFR:MOV:CR 6 6<CR> →	Carriage return terminated command

Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned position	6 6<CR>
Command is complete	← #064<CR>

### 7.3.23 :WFR:NEXT?

**Description:** Move to next testable die and return its location.

**Command Format:** :WFR:NEXT?

**Returned Data:** <col> <row> of stage after the move. If the current die is the last die of the test, the returned position is "2147483647 2147483647"

#### Usage:

Host	P200L_Server
:WFR:NEXT?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned position	7 0<CR>
Command is complete	← #064<CR>

### 7.3.24 :WFR:PATREC

**Description:** Enable/Disable pattern recognition. Video must be running with an active template in order to use pattern recognition.

**Command Format:** :WFR:PATREC <enable>

**Returned Data:** N/A

#### Usage:

Host	P200L_Server
:WFR:PATREC true →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.3.25 :WFR:PATREC?

**Description:** Get Enabled/Disabled state of pattern recognition.

**Command Format:** :WFR: PATREC?

**Returned Data:** <enabled> state

**Usage:**

Host	P200L_Server
:WFR: PATREC?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned state	True<CR>
Command is complete	← #064<CR>

### 7.3.26 :WFR:POS:CR?

**Description:** Get current Column, Row position

**Command Format:** :WFR: POS : CR?

**Returned Data:** <col> <row> position of stage

**Usage:**

Host	P200L_Server
:WFR: POS : CR?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned position	7 0<CR>
Command is complete	← #064<CR>

### 7.3.27 :WFR:REFCR Set the Column, Row position

**Description:** Set the current position to a Column, Row.

**Command Format:** :WFR: REFCR <col> <row>

Where <col> <row> is the new position of stage

**Returned Data:** <x> <y> location of the <col> <row> reference position

**Usage:**

Host	P200L_Server
:WFR:MOV:CR 6 6<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned position	-60.0000 100.0000<CR>
Command is complete	← #064<CR>

## 7.3.28 :WFR:REFCR?

**Description:** Get the X,Y offset (in mm) of the zeroth Column, Row. This offset is different than the user defined X,Y offset available in Navigator. It is referenced to the home location.

**Command Format:** :WFR:REFCR?

**Returned Data:** <cref> <rref> XY offset from home

**Usage:**

Host	P200L_Server
:WFR:MOV:CR 6 6<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned offset in millimeters	-60.0000 100.0000<CR>
Command is complete	← #064<CR>

## 7.3.29 :WFR:RESET

**Description:** Reset the bincodes of active die to “Untested” and clear the map test cursor.

**Command Format:** :WFR:RESET

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:WFR:RESET<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>



Command is complete	← #064<CR>
---------------------	------------

### 7.3.30 :WFR:SUBSITES

**Description:** Enable/Disable Navigator Sub-Site tests at each die.

**Command Format:** :WFR:SUBSITES <enable>

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:WFR:SUBSITES true<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.3.31 :WFR:SUBSITES?

**Description:** Get Enabled/Disabled state of Navigator Sub-Site tests.

**Command Format:** :WFR:SUBSITES?

**Returned Data:** <enable> state

**Usage:**

Host	P200L_Server
:WFR:SUBSITES?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned Sub-Site test flag	True<CR>
Command is complete	← #064<CR>

## 7.4 Video TCP/IP Commands

Comand	Description
:PAT:CAMERA	Set the active camera
:PAT:CAMERA?	Get the active camera
:PAT:CLEARREF	Clears the user defined reference mark
:PAT:CORRECT	Adjust stage X-Y by the correction error
:PAT:CORRECT?	Get measured correction error
:PAT:METRICS	Set the pattern matching threshold
:PAT:METRICS?	Get the pattern matching threshold
:PAT:SHOWREF	Display a reference mark at the indicated offset from the taught pattern
:PAT:TEACH	Teach the template from the current live image.
:PAT:TEMPLATE	Set the template from a file
:PAT:TEMPLATE?	Get the template file

### 7.4.1 :PAT:CAMERA

**Description:** Select the live imaging source.

**Command Format:** :PAT:CAMERA <camera\_index>

**Returned Data:** N/A

#### Usage:

Host	P200L_Server
:PAT:CAMERA 1<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.4.2 :PAT:CAMERA?

**Description:** Get the live imaging source.

**Command Format:** :PAT:CAMERA?

**Returned Data:** <camera\_index> Index of selected camera in the combo box.

#### Usage:

Host	P200L_Server
:PAT:CAMERA?<CR> →	Carriage return terminated command

Indicates server is busy with the command	← #080<CR>
	← #192<CR>
Returned index of selected camera	1<CR>
Command is complete	← #064<CR>

#### 7.4.3 :PAT:CLEARREF

**Description:** Clears the user defined reference mark.

**Command Format:** :PAT:CLEARREF

**Returned Data:** <enable> state

#### Usage:

Host	P200L_Server
:PAT:CLEARREF<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.4.4 :PAT:CORRECT

**Description:** Adjust stage X-Y by the correction error.

**Command Format:** :PAT:CORRECT

**Returned Data:** <xerr> <yerr> <score> <threshold>

Where <xerr> <yerr> is the error offset in mm between the taught template and live image.

<score> is the pattern match score in the range 0.0 to 1.0, with 1.0 being a perfect (100%) match.

<threshold> is the score threshold considered to be a match.

#### Usage:

Host	P200L_Server
:PAT:CORRECT<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>

Returned correction values in mm. Since the score is larger than the threshold, the pattern is considered a match.	0.075000 -0.015000 0.910217 0.530000<CR>
Command is complete	← #064<CR>

#### 7.4.5 :PAT:CORRECT?

**Description:** Get measured correction error in mm, but do not move the stage.

**Command Format:** :PAT:CORRECT?

**Returned Data:** <xerr> <yerr> <score> <threshold>

Where <xerr> <yerr> is the error offset in mm between the taught template and live image.

<score> is the pattern match score in the range 0.0 to 1.0, with 1.0 being a perfect (100%) match.

<threshold> is the score threshold considered to be a match.

#### Usage:

Host	P200L_Server
:PAT:CORRECT?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned correction values. Since the score is larger than the threshold, the pattern is considered a match.	0.075000 -0.015000 0.910217 0.530000<CR>
Command is complete	← #064<CR>

#### 7.4.6 :PAT:METRICS

**Description:** Set the pattern matching threshold.

**Command Format:** :PAT:METRICS <threshold>

Where <threshold> is the pattern match score threshold in the range 0.0 to 1.0, with 1.0 being a perfect (100%) match.

**Returned Data:** N/A

#### Usage:

Host	P200L_Server
:PAT:METRICS 0.5<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.4.7 :PAT:METRICS?

**Description:** Get the pattern matching threshold.

**Command Format:** :PAT:METRICS?

**Returned Data:** <threshold> is the pattern match score threshold in the range 0.0 to 1.0, with 1.0 being a perfect (100%) match.

#### Usage:

Host	P200L_Server
:PAT:METRICS?<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Returned threshold value	0.500000<CR>
Command is complete	← #064<CR>

#### 7.4.8 :PAT:SHOWREF

**Description:** Display a reference mark at the indicated offset from the taught pattern.

**Command Format:** :PAT:SHOWREF <xoff> <yoff> <color | Off>

Where <xoff> <yoff> are the offset in mm from the taught location.

<color> is the name of a Windows color. Common color names include: Black, Brown, Red, Orange, Yellow, Green, Blue, Violet, Gray, White, Cyan. The modifiers Light, Medium, and Dark can be prepended to adjust intensity – e.g. LightBlue. See [https://msdn.microsoft.com/en-us/library/aa358802\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/aa358802(v=vs.85).aspx) for the complete list of colors.

Setting the color to “Off”, turns off the reference mark.

If a color is not recognized, the color defaults to “Black”

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
:PAT:SHOWREF .010 -.033 Aquamarine<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

## 7.4.9 :PAT:TEACH

**Description:** Teach the template from the current live image.

**Command Format:** :PAT:TEACH

**Returned Data:** <valid> indicates if the template is taught successfully. The score threshold may need to be adjusted down if this value is False.

**Usage:**

Host	P200L_Server
:PAT:TEACH<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Template is valid	True<CR>
Command is complete	← #064<CR>

## 7.4.10 :PAT:TEMPLATE

**Description:** Set the template from a file.

**Command Format:** :PAT:TEMPLATE <filename>

Where <filename> is a valid template file. An error is returned if the file cannot be loaded.

**Returned Data:** N/A

**Usage:**

Host	P200L_Server
------	--------------

:PAT:TEMPLATE template_2.png<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

#### 7.4.11 :PAT:TEMPLATE?

**Description:** Get the template file.

**Command Format:** :PAT:TEMPLATE?

**Returned Data:** <filename> of the active template file. If no file is loaded, the returned value is "<none>".

#### Usage:

Host	P200L_Server
:PAT:TEMPLATE<CR> →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data is available	← #192<CR>
Template is valid	True<CR>
Command is complete	← #064<CR>

## 7.5 Router TCP/IP Commands

Command	Description / Reference Command Set
*EX	Terminate Router
*IDN?	Get Router version information
*RST	Reset to default states
:JOY:IL	netProbe 7 P200L Server
:JOY:IL?	netProbe 7 P200L Server
:JOY:NAVLOCK	netProbe 7 P200L Server
:JOY:NAVLOCK?	netProbe 7 P200L Server
:PAT:CAMERA	netProbe 7 Video
:PAT:CAMERA?	netProbe 7 Video
:PAT:CLEARREF	netProbe 7 Video
:PAT:CORRECT	netProbe 7 Video
:PAT:CORRECT?	netProbe 7 Video
:PAT:METRICS	netProbe 7 Video
:PAT:METRICS?	netProbe 7 Video
:PAT:SHOWREF	netProbe 7 Video
:PAT:TEACH	netProbe 7 Video
:PAT:TEMPLATE	netProbe 7 Video
:PAT:TEMPLATE?	netProbe 7 Video
:PRB:AL	netProbe 7 P200L Server
:PRB:AL?	netProbe 7 P200L Server
:PRB:CEN:ALL	netProbe 7 P200L Server
:PRB:CEN:VAL	netProbe 7 P200L Server
:PRB:CEN:VAL?	netProbe 7 P200L Server
:PRB:DN	netProbe 7 P200L Server
:PRB:DNM	netProbe 7 P200L Server
:PRB:DNM?	netProbe 7 P200L Server
:PRB:HO:ALL	netProbe 7 P200L Server
:PRB:HO:ALL?	netProbe 7 P200L Server
:PRB:JOG:XY	netProbe 7 P200L Server
:PRB:JOG:Z	netProbe 7 P200L Server
:PRB:KI	netProbe 7 P200L Server
:PRB:MOV:XYA	netProbe 7 P200L Server
:PRB:MOV:XYNO	netProbe 7 P200L Server
:PRB:MOV:XYRE	netProbe 7 P200L Server
:PRB:MOV:ZA	netProbe 7 P200L Server
:PRB:MOV:ZRE	netProbe 7 P200L Server
:PRB:POS:XYZ?	netProbe 7 P200L Server
:PRB:POS:XYZNO?	netProbe 7 P200L Server
:PRB:POS:XYZR?	netProbe 7 P200L Server



:PRB:REFXY	netProbe 7 P200L Server
:PRB:REFXY?	netProbe 7 P200L Server
:PRB:STO	netProbe 7 P200L Server
:PRB:UDO	netProbe 7 P200L Server
:PRB:UDO?	netProbe 7 P200L Server
:PRB:UN	netProbe 7 P200L Server
:PRB:UN:VAL	netProbe 7 P200L Server
:PRB:UN:VAL?	netProbe 7 P200L Server
:PRB:UP	netProbe 7 P200L Server
:SUB:CURR?	netProbe 7 P200L Server
:SUB:FILE	netProbe 7 P200L Server
:SUB:FILE?	netProbe 7 P200L Server
:SUB:FIRST?	netProbe 7 P200L Server
:SUB:GOTO	netProbe 7 P200L Server
:SUB:NEXT?	netProbe 7 P200L Server
:SUB:PREV?	netProbe 7 P200L Server
:SUB:RESET	netProbe 7 P200L Server
:WFR:AUTOIL	netProbe 7 Wafer Map
:WFR:AUTOIL?	netProbe 7 Wafer Map
:WFR:BIN	netProbe 7 Wafer Map
:WFR:BIN:LIST?	netProbe 7 Wafer Map
:WFR:BIN:VAL	netProbe 7 Wafer Map
:WFR:CURR?	netProbe 7 Wafer Map
:WFR:DIESIZE	netProbe 7 Wafer Map
:WFR:DIESIZE?	netProbe 7 Wafer Map
:WFR:FILE	netProbe 7 Wafer Map
:WFR:FILE?	netProbe 7 Wafer Map
:WFR:FIRST:VAL?	netProbe 7 Wafer Map
:WFR:FIRST?	netProbe 7 Wafer Map
:WFR:MAP:COUNT?	netProbe 7 Wafer Map
:WFR:MAP:CSV	netProbe 7 Wafer Map
:WFR:MAP:CSV?	netProbe 7 Wafer Map
:WFR:MAP:XML	netProbe 7 Wafer Map
:WFR:MAP:XML?	netProbe 7 Wafer Map
:WFR:MOV:CR	netProbe 7 Wafer Map
:WFR:NEXT?	netProbe 7 Wafer Map
:WFR:PATREC	netProbe 7 Wafer Map
:WFR:PATREC?	netProbe 7 Wafer Map
:WFR:POS:CR?	netProbe 7 Wafer Map
:WFR:REFCR	netProbe 7 Wafer Map
:WFR:REFCR?	netProbe 7 Wafer Map
:WFR:RESET	netProbe 7 Wafer Map

:WFR:SUBSITES	netProbe 7 Wafer Map
:WFR:SUBSITES?	netProbe 7 Wafer Map
ERR?	Get most recent error
STA?	Get motion status
UNITS	Set units of measurement to millimeters, microns, or mils
UNITS?	Get the active units of measurement
UPDT	Set the update enabled state and mask
UPDT?	Get the update enabled state mask

### 7.5.1 UNITS

**Description:** Set the current units of measurement for the Router host interface.

**Command Format:** UNITS <unit>

Where unit is mils (1/1000<sup>th</sup> inch), micron (1/10<sup>6</sup> meters), or mm (1/1000<sup>th</sup> meter)

**Returned Data:** N/A

**Usage:**

Host	P300L_Server
UNITS mils →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Command is complete	← #064<CR>

### 7.5.2 UNITS?

**Description:** Get the current units of measurement for the Router host interface.

**Command Format:** UNITS?

**Returned Data:** <unit>

Where unit is mils (1/1000<sup>th</sup> inch), micron (1/10<sup>6</sup> of a meter), or mm (1/1000<sup>th</sup> meter)

**Usage:**

Host	P300L_Server
UNITS? →	Carriage return terminated command
Indicates server is busy with the command	← #080<CR>
Data Available	← #192<CR>
	← mil<CR>
Command is complete	← #064<CR>