

ProbeMagic™

(Version 5.x for Windows® XP)

KORIMA, INC
119 E. Star Of India Lane
Carson, CA 90746

Tel. (310) 532-2222

Fax. (310) 527-0470

Last Update: March,5th ,2004

Table Of Contents

1. Introduction	4
2. Feature Highlights	5
2.1 On Screen Video	5
2.2 Machine Vision	5
2.3 Point-and-Shoot	5
2.4 Map View	5
2.5 Two Point Theta Alignment	6
2.6 ProbeMagic™ Program Modes	6
2.7 Motion Emulation	7
3. Software Installation	8
3.1 To Install on a New Computer or Directory	8
3.2 To Install a New Version over an Existing Version	8
4. Windows® 95 User Interface Basics	9
5. XY Stage Basics	10
5.1 User Coordinate System.....	10
5.2 Physical Coordinate System.....	11
5.3 Stage Reference.....	11
6. Interactive Features	12
6.1 Pull Down Menus	12
6.2 Tool Bar	15
6.3 User Defined Function keys	16
6.4 Joystick	16
6.5 Control Panel	18
6.6 Command Window	20
6.7 Map View (Wafer Map).....	20
6.8 Live Video	23
6.81 Video Navigator	
6.82 On Screen Dimension Measurement	
6.9 Two Point Alignment	25
6.91 Software Theta	
7. Machine Vision	28
7.1 Pattern Recognition	28
7.2 Auto Focus	30
7.3 Auto Align	31
8. User Programs For Automation	32
8.1 Map Programs	32
8.2 Test Site Programs	34
8.3 Script Programs	35

8.31 An Associated Script Program	38	
9. System Integration	40	
9.1 MS Windows® DDE & OLE Automation.....	40	
10. Saving and Restoring Setups.....	44	
11. Configuration and Startup	45	
12. Open Architecture and DLL Interface	47	
13. Motion Emulation Mode	49	
14. Error Compensation		
14.1 Linear Error Correction		50
14.2 Pattern Recognition	51	
15. Trouble Shooting	52	
16. Command Listing	55	
16.1 Interactive Commands	55	
16.2 Batch Commands	70	
17. Add-On Options	77	
17.1 Probe Card Edge Sensor	77	
17.2 Motorized Theta	80	
17.3 Motorized Turret	81	
17.4 Motorized Microscope & Positioner	82	
17.5 KLA Data File Interface.....	87	
17.6 Orbotech Data File Interface.....	90	
17.7 Spectra-Physics Laser	93	
17.8 Truly (ADTech) Data File	97	
17.9 ADTech Data File.....	99	
Appendix I ProbeMagic™ files.....	102	
Appendix II Sample Programs	103	
Appendix III A Sample Listing of PMAGIC.INI	109	
Appendix IV A Sample Listing of LVIDEO.INI.....	117	

1.0 Introduction

ProbeMagic™ is the main control software for KORIMA's automated large area probe stations and laser repair systems. ProbeMagic™, which operates under Microsoft Windows® 95/98 or XP, controls XY stage motion, probe motions, lasers and other accessories.

The ProbeMagic™ software serves three main functions. It:

1. Provides an application oriented user interface.
2. Optimizes the probe station's hardware performance.
3. Integrates with other software or systems. (e.g. the instrument control software or data analysis software, via GPIB, RS232 or Windows® 95's OLE Automation (a.k.a ActiveX)).

ProbeMagic™ provides an application oriented user interface that integrates all subsystems being utilized. The objective of ProbeMagic™ is to free the user from the complexities of the probe station and allow emphasis to be placed on accurately performing essential tests.

ProbeMagic™ has many features aimed at providing such an intuitive user interface. These features include the On Screen Live Video, Map Views, Point and Shoot, Auto Alignment, Auto Focus, Point and Click Program Setup, among others. In addition, ProbeMagic™ also maintains an effective program development environment that helps the user quickly develop user-specific automated processes. Starting from version 4.0 of ProbeMagic™, Machine Vision has also been integrated into the standard software package. This additional feature provides pattern recognition, auto focus and alignment, furthering ProbeMagic's™ capabilities to automate processes.

With large area probing, stage accuracy and flatness is hard to achieve, but essential. ProbeMagic™ has a sophisticated error correction feature that aims to correct and compensate for imperfections that may exist on the XY stage. For a 24" x 24" precision graded aluminum stage with linear optional encoder feedback, it is difficult to attain an accuracy of less than +/- 30 microns without such software compensation. The largest error is attributed to the imperfections of the XY axes in the perpendicular direction. ProbeMagic™ uses 3 scale factors to correct for linear errors on the X and Y axes, as well as for XY orthogonality. Non-linear errors can also be compensated for using ProbeMagic's™ built-in Machine Vision features, such as pattern recognition and auto focus.

As no probe station is an island, in order to get any job done, it has to integrate or interface with other systems. ProbeMagic™ provides a set of interfaces for easy system integration. Remote interfaces, such as RS232 and GPIB (optional), allow ProbeMagic™ to be driven by a remotely located host computer. Program interface features include the MS Windows® 95 OLE Automation (also known as ActiveX) and clipboard, which allows for communication with other Windows® software. Furthermore, ProbeMagic™ has an open architecture, making it capable of expanding its functionality by linking in option DLLs at run-

time. Users can also take advantage of this interface to have ProbeMagic™ link in their option DLLs (dynamic link library) to perform special functions. This modular approach allows KORIMA to provide the user with highly customized software at an off-the-shelf cost.

2.0 Feature Highlights

This section highlights the main features of ProbeMagic™ 4.0. It will help users to become familiar with the terminology used throughout this document.

2.1 On Screen Video

The on screen video simultaneously displays the live video image that appears on the computer screen on a window alongside, with all the control elements. This not only eliminates the need for a separate video monitor, but also provides a more ergonomically appropriate way of viewing the video, as it eliminates the need to attempt to view two monitors concurrently. In addition, this feature opens the door to a variety of additional convenient features, allowing on screen video images to be stored to disk, retrieved and printed. It also makes using as point-and-shoot navigation and making on screen dimension measurements simple.

2.2 Machine Vision

ProbeMagic™ 4.0 integrates the Machine Vision software library from Matrox (Matrox Imaging Library Version 5.0) into its code. This integration gives ProbeMagic™ a large collection of Machine Vision functions from which specific features can be built, such as pattern recognition, auto focus and alignment. This newly added Machine Vision feature is very useful in process automation and error compensation.

2.3 Point-and-Shoot

Point-and-Shoot is a type of video navigation that utilizes the mouse as its main vehicle to achieve its goals. By simply double clicking on the mouse, you can move about the stage easily and quickly. For instance, if you desire to probe a probe pad found in the top left corner of the video screen, you simply need to move the mouse to point to the pad and double click. The stage will then move and position the pad under the probe. Other video navigation features include drag-and-drop and panning. These are later discussed in the Live Video section of the manual.

2.4 Map View

The Point-and-Shoot feature is very intuitive, requiring you to only point and click a mouse to achieve results. However, in semi-conductor probing, magnifications of 100x to 1000x, with a field of view that is only a few hundreds microns across, is typically used. This type of usage asks for more than the above described Point-and-Shoot feature. A global view of the substrate or wafer being observed becomes necessary to travel long distances. For this type of situation, Map View, when used with Point-and-Shoot, provides the perfect solution by giving the user a complete view of the substrate or wafer.

2.5 Theta Alignment

With the traditional micrometer, attaining substrate alignment can be very tricky and time consuming. With ProbeMagic™, on the other hand, aligning a substrate is very simple, requiring only two points. The computer does all of the rest. Starting from ProbeMagic™ version 4.0, pattern recognition can also be used to recognize the alignment marks, resulting in a totally automated alignment process. There are currently two modes for alignment:

Mode 1 By Line:

This mode requires no setup. It only asks that, during alignment, using the computer, you pick two points along a straight horizontal line on the substrate. The computer then makes the necessary adjustments based on the differences in Y coordinates of the points picked. This mode is useful during setups. (See command SETBASE)

Mode 2 By Marks:

This mode requires two steps. The first is the defining of two alignment marks on a well-aligned substrate. The second takes place during alignment with the system moving close to the alignment marks, and then prompting you to line up the cross hair with the marks. Based on your adjustments, the computer calculates the theta offset. This mode is most suitable for use in processes. Starting from version 4.0, this mode of alignment can be further automated with the pattern recognition feature that eliminates the human operator completely from the alignment process. (See command ALIGN)

Note: If the motorized theta option is installed, the computer will turn on the theta motor to correct for the theta offset. If it is not installed, it will rotate the coordinate system. (See section 6.91 on Software Theta)

2.6 The Program Modes

In addition to many interactive features, ProbeMagic™ also boasts numerous features for automation. The software is integrated with three types of programs for you to implement automated and idiot-proof processes.

2.61 Script Program

A script program, in its simplest form, is a list of interactive commands stored in a file. When run, it executes the given commands in sequence. To enhance a script program's functionality, another set of commands called **batch commands** is added. These batch commands provide functions such as data file access, managing user input/output and monitoring program controls such as looping, branching and user variable definition. A script program integrated with batch commands is the most flexible and powerful way to control the probe station.

2.62 Test Site Program.

A script program, although powerful, is more useful to the veteran user, one familiar with the commands necessary for script programming, than the novice because of the complexity in creating such a program. A user-friendlier program is the Test Site Program. With the Test Site Program, you only need to learn the test site locations and then run the Test Site Program. The

software controls all of the necessary movements and finally touches down the probes. A script program can also be integrated into the Test Site Program to perform more complicated operations, such as in cases where a subroutine is necessary once the Test Site Program is at the test site.

2.63 Map Program.

A map program is similar to the wafer map programs utilized on regular wafer probers. It allows you to define probing patterns by pointing and clicking on a map of the panel. Associating a Test Site or script program to it can enhance the functionality of a map program.

Thanks to the object-oriented implementation of the ProbeMagic™ software, all three types of programs can be used together, if desired. This ability to mix and match allows for expanded functionality and usability.

2.7 Motion Emulation

The ProbeMagic™ software is also capable of running in emulation mode. In this mode, the software emulates stage motion and OLE automation functions. This allows you to develop the control software away from the probe station.

3.0 Software Installation

3.1 Installing ProbeMagic™ to a New Computer or Directory

The KORIMA LCD probe station or laser repair systems are shipped with the ProbeMagic™ software preinstalled on the computer. In the case that you need to change computers or hard drives and install ProbeMagic™ using the CD-ROM disk, the following procedure should be followed:

1. Turn on the computer.
2. Insert the ProbeMagic™ Installation CD into the CD-ROM drive.
3. From Windows® 98/NT's **Start** Menu, select **Run**.
4. Type **x:\ProbeMagic™ Setup\cd_rom\Setup.exe**. Then hit the Enter Key. (where x is the CD-ROM drive letter)
5. Follow the prompts of the installation program.

3.2 Re-installing ProbeMagic™ over an Existing Version

You may need to re-install ProbeMagic™ during software upgrades or when you purchase additional options after the initial installation of the machine. You may choose to backup the existing software to a different directory before re-installing. The procedure for this is similar to installing ProbeMagic™ to a new computer or directory:

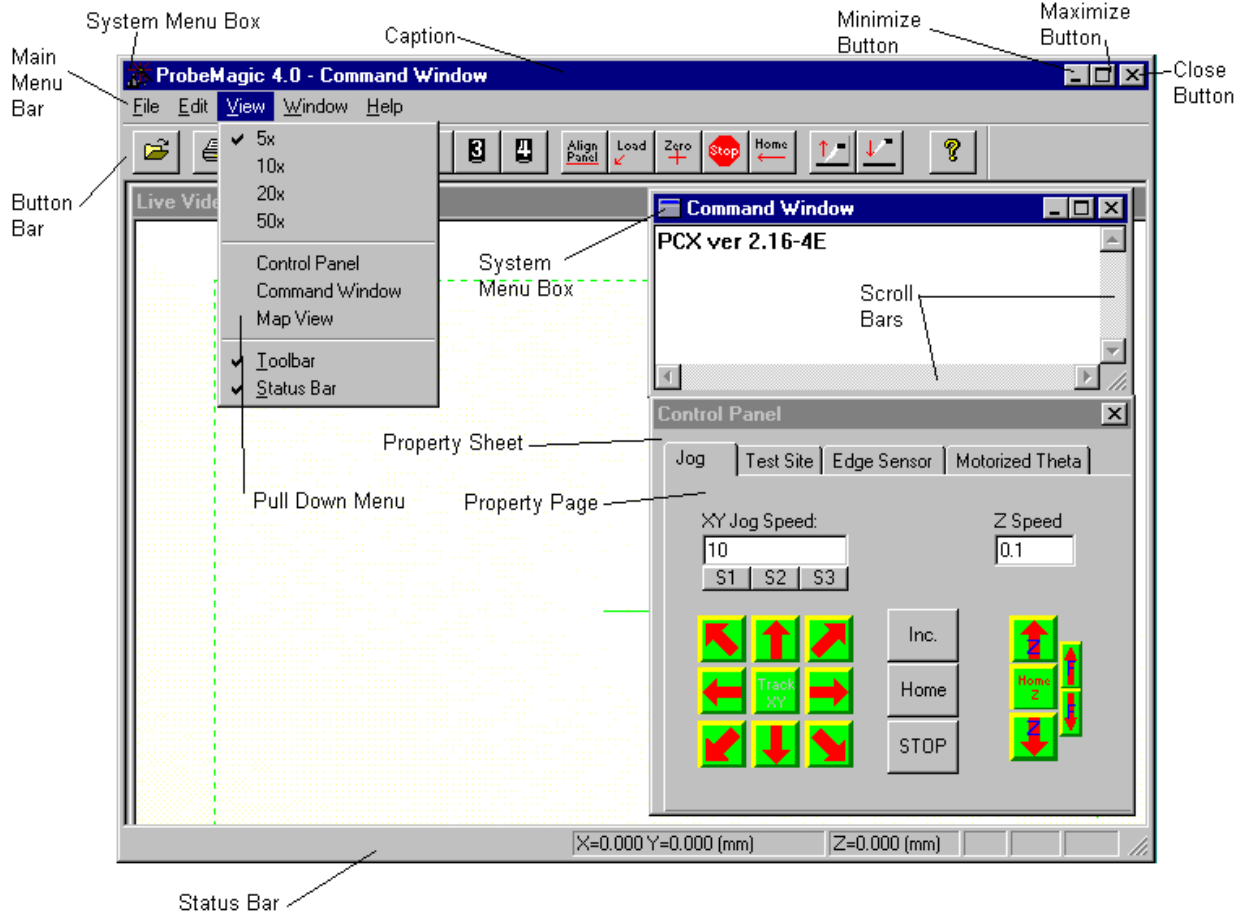
1. Turn on the computer.
2. Insert the installation CD into CD-ROM drive.
3. From the Windows® 98/NT's **Start** Menu, select **Run**.
4. Type **x:\ProbeMagic™ Setup\cd_rom\Setup.exe** and the Enter key. (where x is the CD ROM drive letter)
5. Follow the prompts of the installation program.

When you re-install a new version or add new options, you have to put them in the same directory as that of the previous version. Thus, when the installation program prompts you for a directory, the same directory as that of the previous software must be selected and the **overwrite** option must be selected.

Note: The **overwrite** option will NOT erase any programs already written and saved. All script programs, Test Site Programs and Map Programs will be preserved.

4.0 Windows 95 User Interface Basic

In this section, the basic terminology of the Windows® 95 user interface is introduced. If you are familiar with Microsoft Windows® 95's basic operations, you can skip this section.



4.1 Basic operations:

Select Menu Items

To select a menu item, click on the text on the menu bar. A menu will then pull down, allowing you to select the desired item by clicking on the left mouse button.

Moving a Window

To move a window, point the mouse to the window's caption bar, depress the left mouse button and move the mouse.

Sizing a Window

Only windows with a thick border can be re-sized. To change the size of a window, point the mouse to the border and the mouse pointer will change shape. Then, depress the left mouse button and move the mouse to re-size the window to the desired dimensions.

Closing a Window

To close a window, click on its system menu box. A menu will pull down and select close.

Shrinking a window to an icon

This operation is called minimizing. To do this, click on the minimize button of the window.

Expanding a window to full screen

This operation is also known as maximizing a window. Simply click on the maximize button on the window.

Button Bar

The button bar is the bar under the menu bar. It contains buttons which, when clicked, start certain actions.

Status Bar

This bar displays status information such as current XY stage positions, current scale units, etc. This bar also displays hint text (help information) of menu items and button bar buttons.

Property Sheet and Property Page

The Property Sheet is a collection of pages that are specialized windows that are used as the parent windows for the controls. Examples of controls are buttons, list boxes, check boxes, etc. The Control Panel is a Property Sheet that contains such Property Pages as Jog, the Test Site, etc.

5.0 XY Stage Basics

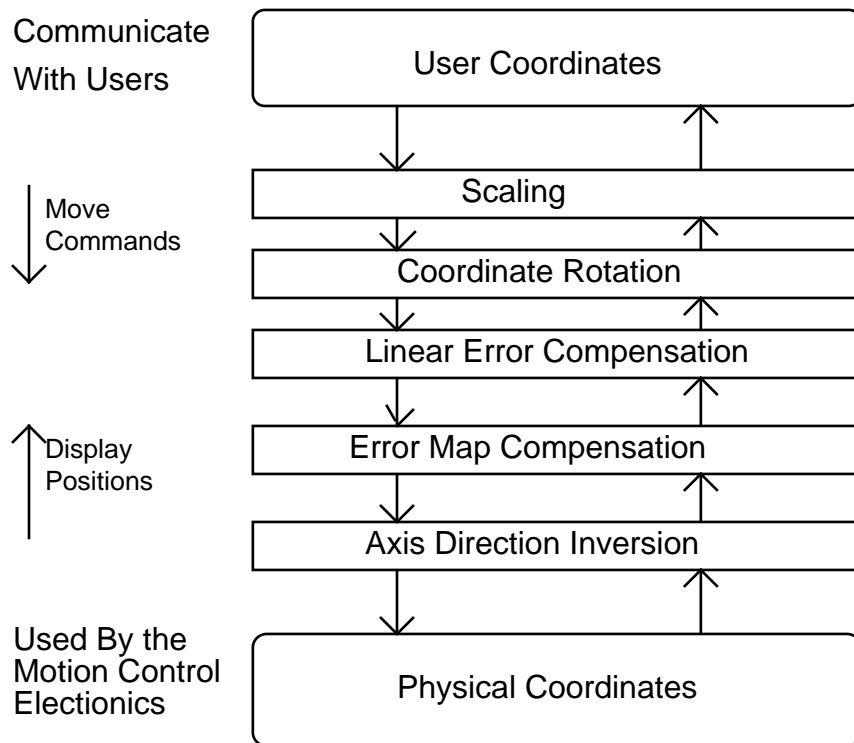
In a LCD probe station or laser repair system, there is an XY stage. The stage travels the X and Y axes to position the substrate under the probes for probing or under the microscope for laser repair. In this section, the coordinate systems used and some of the terminology related to the XY stage will be discussed.

5.1 User Coordinate System

The user coordinate system is the coordinate system that you use to identify the position on the substrate. It is also the only coordinate system that you need to work with. The coordinate system can be used with different scale units (um,mm,mil or inches). The home position (the zero reference position) for these coordinate systems can be designated anywhere on the stage by using the command ZERO. At any given time, the current location of the user coordinate is displayed on the Status Bar for easy reference. It is important that you make sure that the user coordinate system matches with the coordinate system used on the substrate.

5.2 Physical Coordinate System

This is the coordinate system that the motion control electronics uses. Its orientation is fixed by the mechanical location of the stage and the wiring of the motors. The ProbeMagic™ software translates a user coordinate to a physical coordinate before sending it to the motion control electronics. The following drawing shows the relationship between the two coordinate systems.



5.3 Stage Reference

Stage reference helps ProbeMagic™ calculate the absolute position of the stage. Because the optical encoders that track the stage positions are incremental, they only track the number of microns that the stage travels. It cannot decipher the absolute position of the stage. Hence, in order to establish an absolute position reference, the limit switches position of the stage is used. The stage reference is the position at which both axes are at their most negative extremes. You can set the stage reference by moving the stage to the most top and right position (by pressing the south pointing arrow button followed by the west pointing arrow in the Jog page on the Control Panel), and then selecting **Edit, Set Stage Reference** from the menu bar.

You do not need to set the stage reference very often because ProbeMagic™ saves the stage reference position as well as the current position to disk every time you exit the program. Then, when ProbeMagic™ is re-started, it will automatically restore the electronic counters with

the saved information. It is imperative that you exit ProbeMagic™ properly because of this feature. If you experience a power failure, program crash or any unexpected termination of ProbeMagic™, it will become necessary to reset your stage reference.

6.0 Interactive Features

ProbeMagic™ uses pull down menus and dialog boxes to create a user-friendly environment. The user interface is a Microsoft Windows® Multiple Document Interface (MDI). With many similarities to the standard Microsoft interface, Windows® users should find the ProbeMagic™ user interface very familiar. Feel free to explore the ProbeMagic™ software, and experiment with the menus and control buttons.

6.1 The pull down menus

The ProbeMagic™ menu consists of five basic selections. Each selection is comprised of a pull down menu.

File

This pull down menu allows you to manipulate the ProbeMagic™ program files. This menu will change depending upon the active window. The basic selections are:

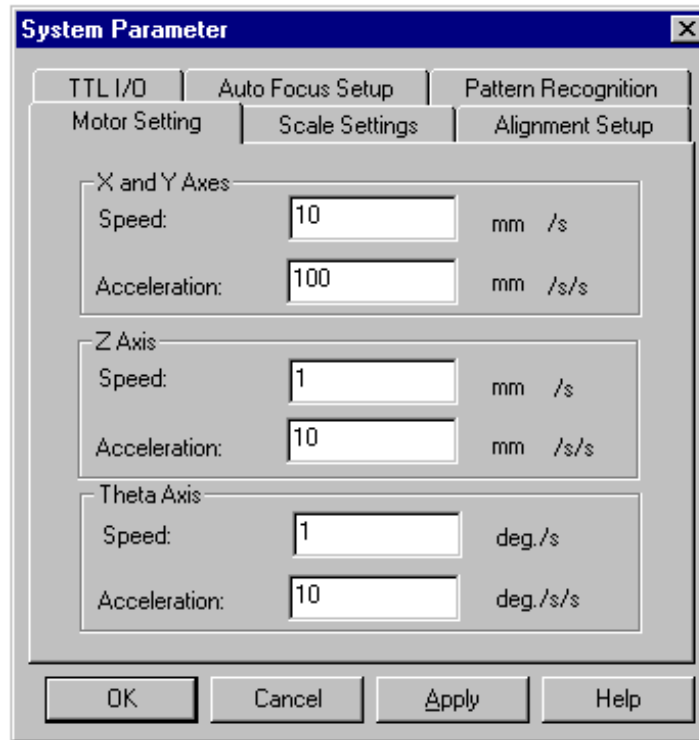
- | | |
|-------------------|---|
| New Script | - Launches the program editor to allow the user to create new script files |
| Open | - Opens the program editor with an existing script file, video image or ProbeMagic™ Setup |
| Save As | - Saves script files, video images and system setup |
| Run Script | - Executes a ProbeMagic™ script program |
| Exit | - Exits the ProbeMagic™ program |

Edit

This selection pulls down the edit menu and allows you to change system settings.

Parameters

- Launches the parameter setup property sheet



Video Setup

Calibrate Navigator

- Calibrates the video Point-and-Shoot feature

Set Par Center Offset

- Launches a process to calibrate potential offsets of objectives (5x, 10x, 20x, etc.) that may not be looking at the exact same place on the wafer

Move Cross Hair

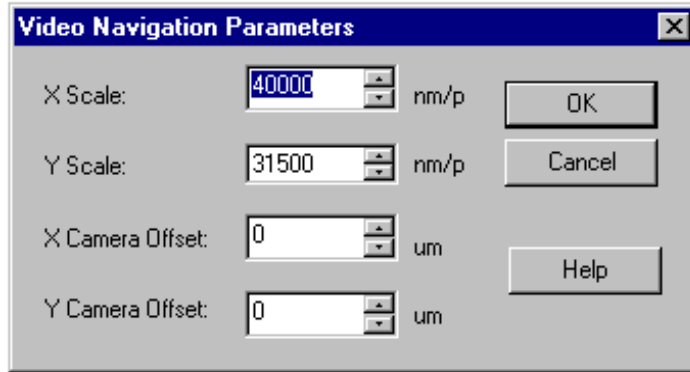
- Repositions the video cross hair

Display Setting

- Changes the video display brightness and contrast

Edit Parameters

- Launches the following dialog for video navigator parameters (see figure below)



Set Stage Reference - Sets the XY Stage's reference position.

View

5x - Changes to 5x Objective

10x - Changes to 10x Objective

20x - Changes to 20x Objective

50x -Changes to 50x Objective

Control Panel - Launches or highlights the Control Panel

Command Window - Launches or highlights the Command Window.

Map View - Launches or highlights the Map View Window

Tool Bar - Shows or Hides the Tool Bar

Status Bar - Shows or Hides the Status Bar

Window

Using this menu, you can bring the selected window to view, above all of the other windows, and make it the new active window. This pull down menu dynamically grows or shrinks depending on the number of windows currently open. The menu selection helps you to locate a window on the screen, making it highly useful when many windows are open concurrently.

Help


From this menu, you are able to access the on-line help information of ProbeMagic™.

6.2 Tool Bar



There are three types of buttons on the tool bar:

1. System Buttons

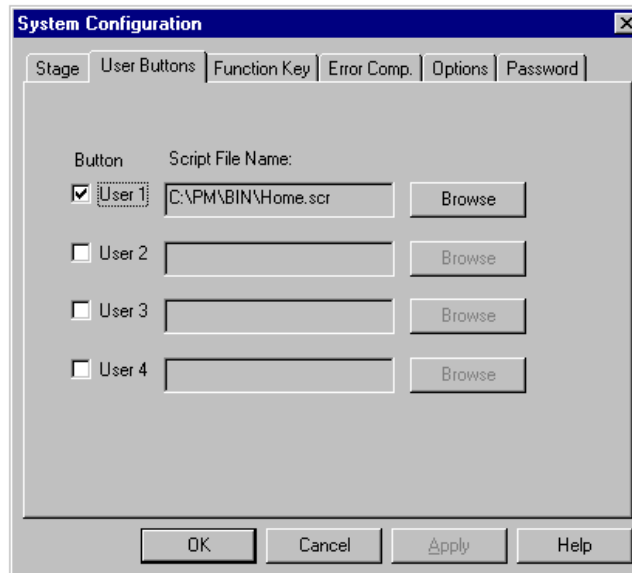
The system buttons are the 18 left most buttons on the button bar. These buttons provide quick access to certain menu selections. For example, by pressing the  button, the Command Inputs window is launched. This is the same as selecting Command Window from the menu bar **View**.

2. Option DLL Added Buttons

These buttons are added by the option dynamic link libraries (DLL) that ProbeMagic™ links into its program. The DLL buttons are located immediately after the system buttons. Depending on the number of DLLs loaded, users may have different numbers of DLL added buttons.

3. User Defined Buttons

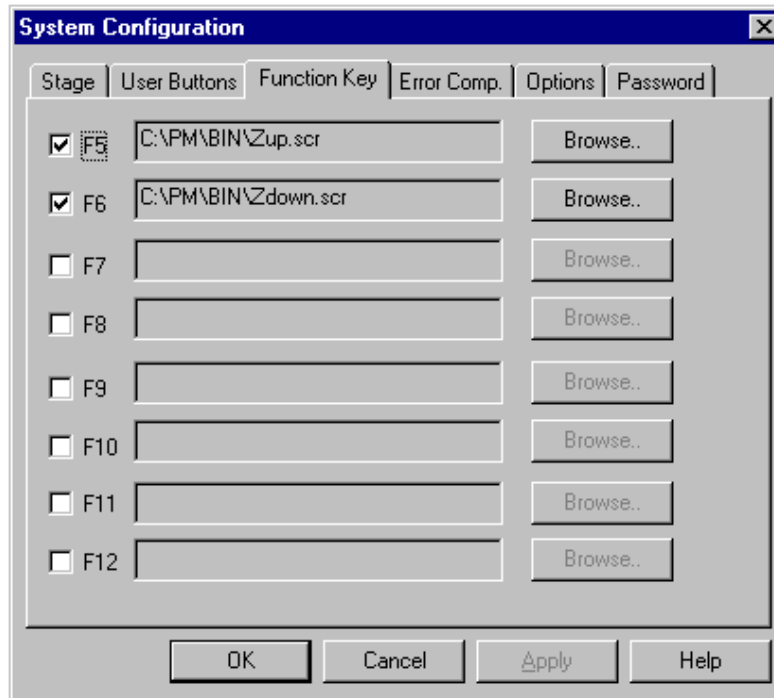
You can define up to 4 buttons on the button bar to do commonly performed actions specific to your applications. A user-defined button will have an associated script program, causing the script file to be executed when the button is pressed.



To add a user button to the button bar, you will first need to create a script file that performs the desired function. For example, if you want to create a button that executes the command ZMV, 0.1, a file that contains the line, ZMC, 0.1 needs to be created. This must then be saved under some name. Lastly, the **System Configuration** Property Sheet, **User Buttons** page must be launched.

6.3 User Defined Function Keys

Function keys, from F5 to F12, can be defined to perform actions specific to your applications. A user defined Function key will have an associated script program, causing the script file to be executed when the Function Key is pressed.



To define a function key action, the steps necessary are similar to those undertaken to define a user button. You first need to create a script file that performs the function. Then, you need to save it and launch the **System Configuration** Property Sheet, **Function Key** page.

6.4 Joystick

ProbeMagic™ is compatible with any PC game port joystick. However, the 4 buttons, 2 axes type joystick is most suited for ProbeMagic's™ applications. When the option of using a joystick is enabled, the joystick provides convenient jog control. The jog motion speed is proportional to the stick deflection. The first button on the joystick (usually the trigger button) serves as the speed range selector when depressed. But, when it is NOT depressed, the joystick's maximum speed is set by the S1 speed on the control panel. With button 1 depressed, the maximum joystick speed becomes S2 speed.

Script files can be used to define the actions of joystick buttons 2 to 4. When you assign script files to function keys F9, F11 and F12, you also assign them to joystick button 2 to 4 by default.

You can write the script files in such a way that the joystick motion calls different subroutines in the script file. The subroutine names “N”, “NE”, “E”, “SE”, “S”, “SW”, “W” and “NW” can be used to assign subroutines to joystick stick deflection. For example, if the following script is assigned to the F9 key:

```

SUB,N
MR,0,10
RET

SUB,E
MR,10,0
RET

SUB,S
MR,0,-10
RET

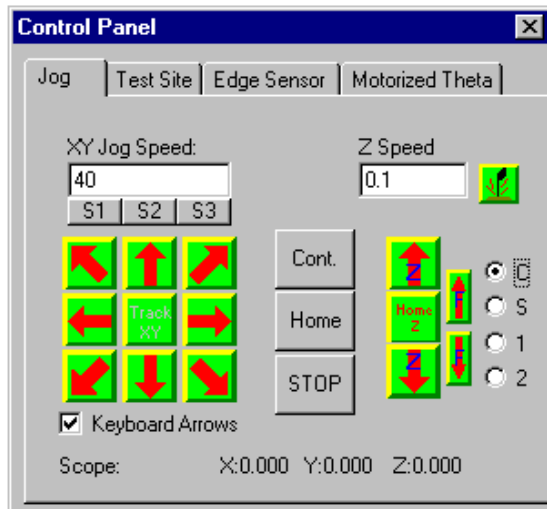
SUB,W
MR,-10,0
RET

BEGIN
MV,0,0
END


```

ProbeMagic™ will execute MV,0,0 when you press the joystick’s 2nd button. If you deflect the stick in a northward direction with the 2nd button depressed, SUB, N will be executed. If the 2nd button is constantly depressed and the stick is moved to the right, the SUB, E subroutine will be executed.

You do not need to define all the subroutines to cover all possible stick directions. If the joystick moves in a direction with no corresponding subroutine, nothing will happen. You may also leave the main routine empty (with END following BEGIN) so that no command will be executed when the button is initially pressed.



6.5 Control Panel

The Control Panel is a property sheet that can consist of up to four pages. They are the Jog page, Test Site page, Edge Sensor page and Motorized Theta page. Using the Jog page, you can move the stage manually by using a trackball (or a mouse). When the control panel is not displayed on the screen, the control panel can be launched by pressing the  button on the button bar.

The Big Arrow Buttons

By pressing the arrow button using the trackball (or mouse), you can move the XY stage. There are two modes of jogging: **Incremental** or **Continuous**. In the **Incremental** mode, each press of the arrow results in a movement of finite length determined by the X and Y Step size settings. In the **Continuous** mode, the channel will move continuously in the selected direction until the button is released.

The Track Buttons

Once this button is selected, the machine enters the track mode. In track mode, axes will track the trackball (or mouse) movement until the right trackball (or mouse) button is clicked. If the left mouse button is depressed while in track mode, the stage will move continuously until the button is released.

The Inc/Cont Button

This button toggles between **Incremental** and **Continuous** modes.

The Home Button

This button moves the active channel to the (0,0) position.

The Stop Button

This button will abort a movement.

By pressing this button, the program being executed will be interrupted, causing a dialog box to pop up that gives you the option to either continue or abort the script file being executed.

The Focus Button

This button moves the focus motor on the microscope to allow focusing of the microscope. Every click on the buttons moves the focus motor a certain number of steps. The step size changes with the objective magnifications. The four step sizes that correspond to the four objectives can be changed by editing the PMAGIC.INI file.

Probe Card Edge Sensor Button

The Probe Card Edge Sensor is a contact-sensing device that ProbeMagic™ uses. An edge sensor detects the contact position during a probe touch down. When the Edge Sensor option is installed, this button toggles the arm state of the edge sensor. You can arm and disarm the edge sensor by clicking on this button.

The Test Site page is explained in section 8.2 Test Site Program.

The Edge Sensor page is explained in section 17.2 Probe Card Edge Sensor.

The Motorized Theta page is explained in section 17.3 Motorized Theta.

Keyboard Arrow Keys

By default, the keyboard interface is enabled to allow you to activate the motion arrows and buttons on the Control Panel by pressing certain keys on the keyboard. Here is the key assignment:

The four arrow keys correspond to the four arrow buttons on the panel.

The **Insert** key presses the  button.

The **Delete** key presses the  button

The **Home** key presses the  button.

The **End** key presses the Z Home button.

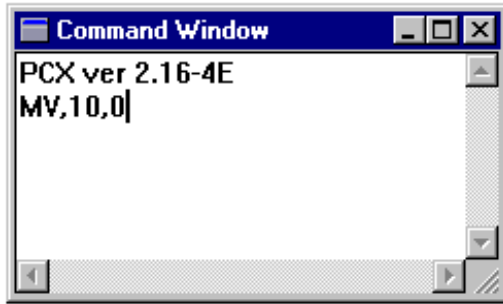
The **Page Up** and **Page Down** keys press the two Z axis arrows

Normally, the arrow key jog function is not active whenever the input focus is in an edit window such as Command Window or Script editor. A script command, KEYFOCUS, can be used to direct the input focus from a script file. Also if the INI file setting:

```
[Control]
ArrowKeyAlways=
```

is set to one, the arrow key jog function is set to always be active.

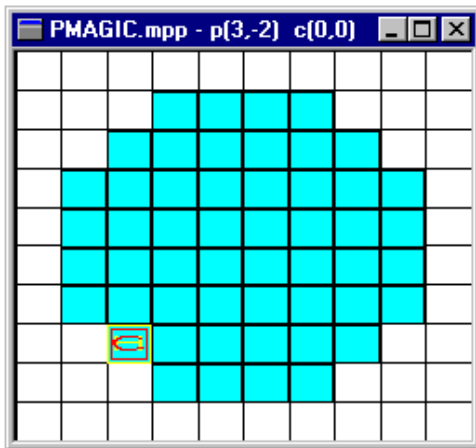
6.6 Command Window




From this window, you can execute interactive commands by typing them via the keyboard. For example, you may type MV,10,12 to command the current channel to move to the location (10,12).

Context sensitive on-line help is available on each command. To use the help feature, simply move the text cursor to a line containing the command and press **F1**. This will cause the related help information to be displayed. For example, to get help information on the MV command, move the text cursor (I beam cursor) using the up and down arrow keys on the keyboard to the line MV,10,10. Then press the **F1** key. Help information on the commands can also be accessed via the Help Index system by selecting **Help** from the menu bar.

6.7 Map View



Map View gives a global view of the substrate. You can move the XY stage by pointing at the map and double clicking. The stage will move to wherever on the panel/wafer you point at. If the map view is not displayed on the screen, you can launch the window by pressing the  button on the button bar.

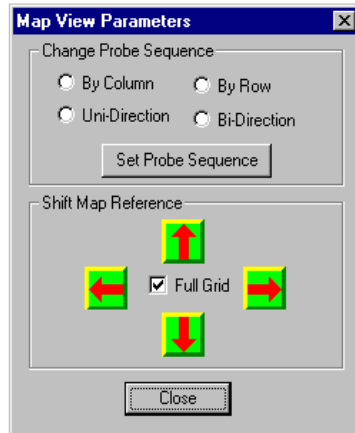
In its local popup menu, the Map View window has the following selections:

Goto Die/Grid Home - Home to Die Home

- Set Map Reference** - Sets the bottom left corner of the map. (Map Reference)
- Run Map Program** - Executes the map program.
- Select Die to Probe** - Toggles die selecting mode, to select die to probe in the map program
- Un-select Die** - Toggles the de-select die mode (This mode de-selects probe dice.)
- Load Map Program** - Loads a map program from disk file

Save Map Program - Saves the current map program to disk file

Set Parameter - Launches the following dialog:



Set Probe Sequence - Rearranges the sequence of motions based on user selection

Shift Map Reference - Changes the map reference incrementally to match the map position with the wafer/panel position.

Edit Data as Text

Here is a listing of the map parameters. All dimensions are in microns.

```
[Map0]
BoardSizeX=100000
BoardSizeY=100000
DieSizeX=10000
DieSizeY=10000
DieCornerX=4499.51
DieCornerY=9502.38
RefPosX=-24125
RefPosY=-29892.9
GridOnFlag=1
SnapToGrid=1
FixDieSize=1
DisplayHomeDie=1
CDieColor=255
HDieColor=65535
DwellTime=1000
AssoFile=
```

```
[DieList]
0 0 0
1 0 0
```

..
..

[DieListEnd]

Descriptions of the parameters that you may edit:

- DieSize** - The X and Y grid size of the map in micron.
- SubstrateSize** - The total size of the map (usually the panel size) in micron.
- GridOnFlag** - Turns the grid On or Off, 1 - On, 0 -Off
- DwellTime** - Signifies wait time in ms before moving to next die in a map program. This parameter is ignored if an associated script file or a sub map is used.
- AssoFile** - Specifies a script program or test site program as an associated file to the map program.

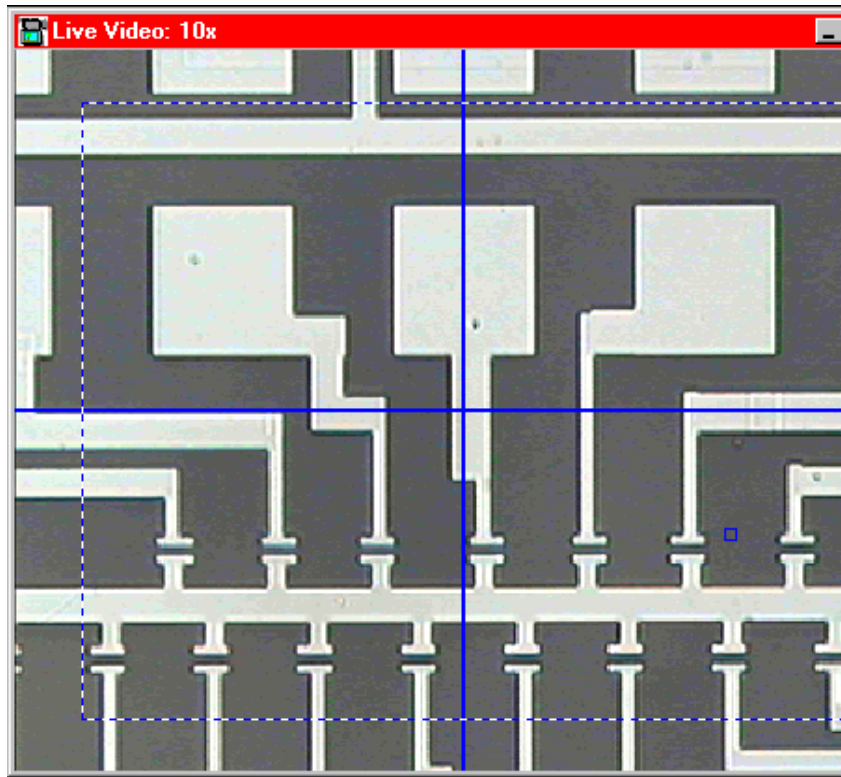
Steps to create a map view:

1. Select from the pop up menu **select, Edit Data as text**, and set Substrate size (panel size) and the Die Size (Grid Size), then close the editor window.
2. Put the panel on the XY stage.
3. Align the panel.
4. Move to the bottom left corner of the panel. Then select **Set Map Reference**.

Please read Section 8.1 on creating a map program.

6.8 Live Video

The live video feature displays the video images generated by the video camera on the computer's VGA monitor. This display can be full screen or in a window alongside the control dialog boxes and menus. By putting two screens in one, you have an integrated control environment for easy operation.



With Live Video, you can capture and save images on the computer's hard disk for later retrieval or print out. Images saved can be printed using an ordinary printer, (preferably a laser printer such as HP Laser Jet®). This feature can also generate a cross hair that eliminates the need to use a separate cross hair generator for targeting.

All of the video attributes such as brightness, contrast, color etc. can be adjusted. Video settings are saved in the file LVIDEO.INI and are automatically set upon program startup.

6.81 Video Navigator

The video window is much more than just a video image on a computer screen. It is a powerful tool to allow movement. For example, if you want to position the stage so that a selected image feature (e.g. a probe pad) will line up with the cross hair, you just need to use the mouse to point to that feature, and then double click on the left mouse button. (This is an

example of the Point-and-Shoot feature.) If you want to move the feature to a position other than the center of the screen, you can use the mouse to drag (pointing to the feature while holding down the left mouse button and moving) the feature to that position. (This is an example of the Drag-and-Drop feature). If the feature of interest is outside the viewing area, you can "pan" the video window. To do so, hold down the left mouse button while inside the video window and then move the mouse pointer outside. Once the mouse pointer is outside the video window, the stage will start to move towards the mouse pointer. The farther the mouse pointer is from the video window, the faster the stage will move. The moving speed begins at the slowest jog speed, S1.

In order for the video navigator to function correctly, it must be properly calibrated. The calibration procedure determines screen magnification (pixel/micron). To calibrate the video navigator, select from the menu **Edit, Video Setup, Calibrate Video Navigator**. You will be prompted to move the stage so that an image feature lines up with the cross hair. After hitting the OK button, you will be asked to move the stage so that the feature moves towards the bottom right corner of the video window and lines up with the newly positioned cross hair. Hit OK when you have lined it up. In the last step, you are asked to confirm your action. You will need to perform this calibration procedure 4 times, once for each objective magnification. After the calibration, the software saves 4 sets of scale factors in the file LVIDEO.INI.

In order for the calibration to stay true, you must not change the zoom setting of the microscope after calibration. We recommend users to keep the zoom ratio at 1 during both calibration and operation. Changing the size of the video window will not affect the video navigator. The video navigator automatically compensates for the size of the video window and for different objective lens on the turret.

ProbeMagic™ can be setup to automatically par center and par focus the microscope. When a microscope is par centered, changing magnification by rotating the objectives will not cause the field of view to shift. When a microscope is par focused, changing magnification by rotating the objectives will not cause the focus to shift. To calibrate these features, select from the menu, **Edit, Video Setup, Set Par Center Offsets**. The prompts will guide you through the process.

6.82 On Screen Dimension Measurement

After the Video Navigator is calibrated, you can adjust the on screen dimension measurements. To enter the measurement mode, click the right mouse button while in the video window. A floating menu will show. Here, select **Measure Distance** and the mouse pointer will change to a cross.

To measure, point the mouse to the starting point of your measurement, depress the left mouse button, and then drag the mouse to the ending point. The measured dimension is displayed in microns below the mouse pointer.

6.9 Two Point Alignment

The mechanical theta adjustment is achieved by having a rotary plate pivot at the center and a micrometer adjust its angle. The purpose of the alignment process is to place the substrate so that the horizontal lines on the substrate run parallel with the X axis of the stage. If done manually, you would have to move the stage back and forth (left and right) and adjust the micrometer while looking at the video image of the substrate until the horizontal lines on the substrate no longer shift up and down as the stage moves back and forth.

With a motorized stage, there is a more efficient way to calculate the misalignment:

1. The first mode of calculating the misalignment is called **Align by Line**. A straight line is picked on the substrate and the XY positions of the left and right most points on the line are recorded. With the coordinates of these two points, the angular misalignment can be easily calculated using simple trigonometry. The following equation gives theta:

$$\text{Theta} = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

With this method, the software will prompt you to pick the two points on a straight line, which can be easily determined with the On Screen Video and Video Navigator features. Then, the above calculation needs to be done. With the theta offset determined, you need only to make the adjustment for the offset. (See SETBASE command.)

2. The second mode is called **Align by Marks**

Alignment marks are frequently added to a substrate to facilitate the alignment process. This mode of alignment takes advantage of these alignment marks. This mode has a setup phase during which you tell the software where the ideal positions of the alignment marks are. The ideal positions are at the positions where the substrate and marks are aligned. (To start the setup process, align a panel using SETBASE, then select from the menu bar, **Edit, Parameters, Alignment** page.) If you know the nominal positions of the alignment marks, you can just enter them into the boxes provided. If the nominal positions aren't known, you can press the **Learn** button and the system will display dialogs to prompt you to move to each alignment mark. During the alignment operation, the machine will move to the alignment marks and prompt you to line up the video cross hair with the alignment marks. Based on the adjustments made and the ideal positions of the alignment marks, the software will calculate the theta offset and make the adjustment. (See ALIGN command) If the Auto Align is enabled, after being taught, the vision will take the place of the operator and line up the video cross hair with the alignment marks. (See section 7.3 Auto Alignment) With vision, the alignment process becomes totally automatic.

6.91 Making Theta Adjustments

After the theta offset is determined, there are two ways that ProbeMagic™ corrects the theta offset. If the motorized theta option is installed, ProbeMagic™ will move the theta motor to correct for the theta offset. You can think of the theta motor as a motorized micrometer for theta. The motor is a stepper motor that can move the precise number of steps to correct for the theta offset. Since there is some mechanical backlash in the motorized theta assembly, the motor will compensate for it by moving beyond the required number of steps in one direction and moving back in that same direction.

Also, due to the backlash on the theta assembly, the system may need to be moved back and forth twice to get the alignment correct, meaning, undergoing the alignment process twice. However, if it becomes necessary for you to align more than twice, it could mean that you have a wrong scale factor for the theta motor. (The scale factor can be changed by editing the PMAGIC.INI file.)

6.92 Software Theta

Software theta is a technique that mathematically rotates the user coordinate system to compensate for the misalignment of the substrate. There is no mechanical movement, thus no special options need to be purchased and no backlash problems need to be compensated for. In some cases, software theta is more desirable than the motorized theta.

For example, if there is a 5 degree offset in theta for the substrate and you want to go to (1000,1000) on the substrate, you need only to type MOVE, 1000, 1000. The software will then calculate the corresponding stage coordinates using the following algorithm:

$$\begin{aligned} X' &= X \cos(A) + Y \sin(A) \\ Y' &= -X \sin(A) + Y \cos(A) \end{aligned}$$

In this case, the figures are:

$$\begin{aligned} X' &= 1000 * \cos(5) + 1000 * \sin(5) \\ Y' &= -1000 * \sin(5) + 1000 * \cos(5) \end{aligned}$$

$$\begin{aligned} X' &= 1083.35 \\ Y' &= 909.04 \end{aligned}$$

The software then commands the stage to go to (1083.35,909.04). This is the same as (1000,1000) on the substrate. The reverse calculation is performed when displaying the coordinates, and thus the Status bar shows (1000,1000). The speed in jog mode is also adjusted as necessary. Linear interpolation is used to allow a straight-line jog motion even with an angular offset.

The software theta will not work in applications that require the probes to be separated far apart from each other, such as in the Open/Short test, where the probes are on two ends of a row or column line. The software theta works well for laser repair or inspection applications.

7.0 Machine Vision

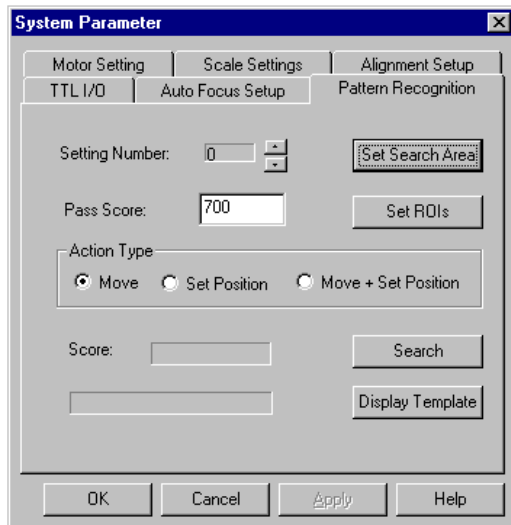
Starting from version 4.0, ProbeMagic™ integrates the Machine Vision library (MIL Version 5.0) from Matrox into its core software. The library functions utilize Intel's MMX® technology to execute sophisticated Machine Vision algorithms on Pentium CPUs without specialized hardware DSPs. The combination of an extensive Machine Vision library and MMX technology produces an extremely versatile and high performance machine. The ProbeMagic™ software applies the Machine Vision function in the following three applications: Pattern Recognition, Auto Focus and Auto Alignment.

7.1 Pattern Recognition

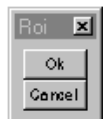
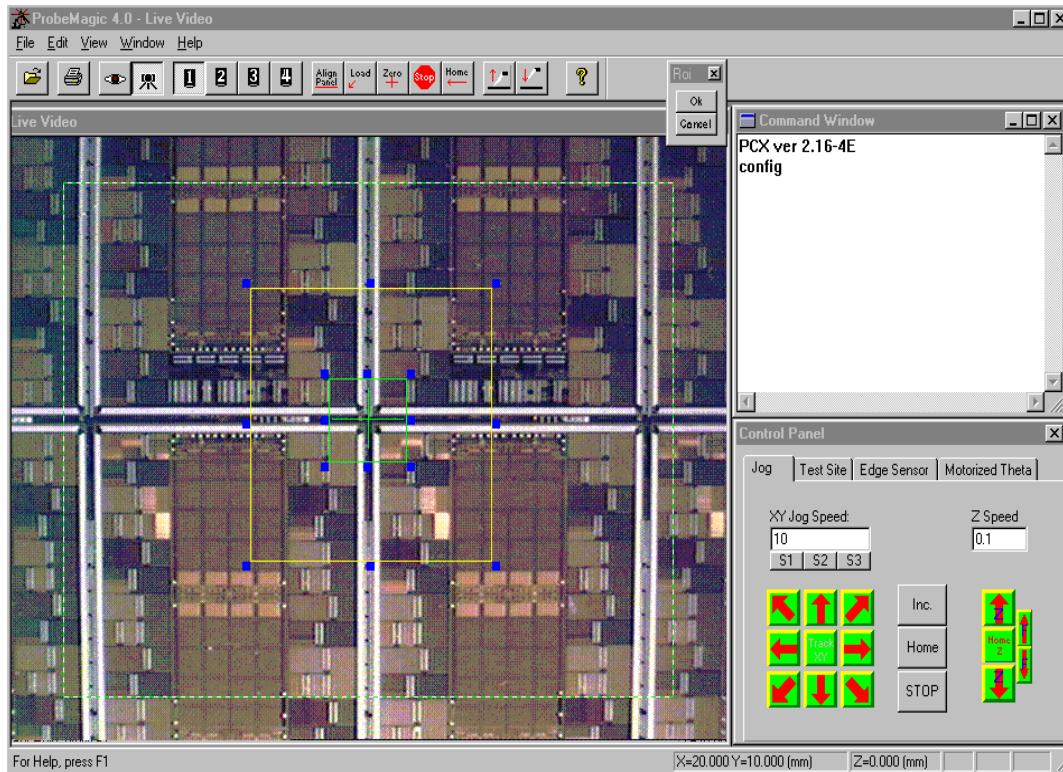
Pattern recognition involves locating the position of previous learned patterns (templates). The techniques are widely used in factory automation applications. With ProbeMagic™, it can be used to compensate for XY stage positioning errors and auto alignment.

To use pattern recognition, a known good pattern must be learned. This process is called template training. While training the template, the ideal screen position of the template is also recorded. A “search area” is then specified. The search area is a region of the screen within which the template image is expected to be found.

During recognition, the machine searches within the designated search area and tries to locate the template image. The algorithm used to match the template image with the scene is called a normalized gray scale correlation. The computation produces a score that indicates how similar the two images are. In ProbeMagic™, a score of 1000 indicates a perfect match. But, in the real world, no two images are exactly alike. Hence, the software allows you to specify the acceptable degree of similarity by setting a Pass Score. A pass score of 700 is usually a good default.



ProbeMagic™ allows you to setup 10 (0 to 9) different sets of parameters for pattern recognition. To set up the pattern recognition, select **Edit, Parameters** from the main menu. The **System Parameter** property sheet will display, and select the **Pattern Recognition** page. The **Setting Number** selects which of the 10 parameter sets you are setting. The **Pass Score** specifies the matching acceptance. The **Set ROI** button begins the template training process. By pressing it, two ROI (region of interest) boxes will appear in the video window.



The green box specifies the template, while the yellow box specifies the search area. The boxes can be sized and moved with the mouse. After both ROIs are specified, press the **OK** button in the ROI dialog box.

The **Action Type** specifies the type of action that the machine performs after locating the pattern. It can be either the **Move** or **Set Position**. For example, if the machine finds a template at (10,11), and the ideal position of the template is (10,10), and the action type specified is **Move**, a move of (0,-1) will be made. If **Set Position** is the action specified, the current position (x,y) will be set to (x,y-1). In addition, the **Search** button allows you to test and search the pattern in the current field of view. The **Display Template** button displays a separate window showing the template trained.

The pattern recognition function can be accessed from a script file using the command:

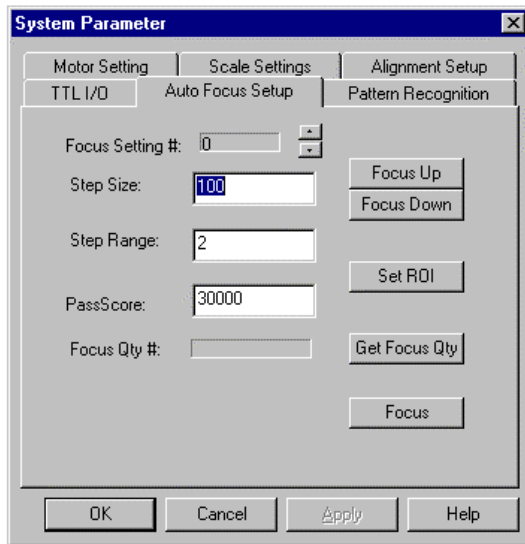
PATREC,p

Where p is the parameter set to a number (0 to 9)

7.2 Auto Focus

Auto focus analyzes image characteristics to determine image focus. The gray level difference between adjacent image pixels is greater in a sharply focused image than in a blurred image. The software computes the “Robert’s cross gradient” for all the pixels in the specified area to determine focus quality. The system drives the focus motor up and down, measuring focus quality at each position to determine the best focus.

ProbeMagic™ allows you to setup 10 sets of parameters for auto focus. To set up auto focus, select **Edit, Parameters** from the main menu. The **System Parameter** property sheet will display, and from there, select the **Auto Focus** page.



The **Focus Setting #** specifies which of the 10 settings you desire. The **Step Size** is the size of the focus distance in um. The **Step Range** indicates the system’s movement in each direction when searching for the best focus quality. A step range of 2 and step size of 100 means the system will go up 200um to take a focus measurement and move down in 4 total steps every 100um to take a focus measurement at each stop.

The **Set ROI** button sets the area for which focus quality is measured. You can also set a **Pass Score** for focus quality. The focusing process will stop when the pass score is reached. The **Focus Up** and **Focus Down** buttons move the focus motor up or down by the Step Size. The **Get Focus Qty** checks the focus quality with the current image. The **Focus** button exercises the complete auto focus process.

The auto focus function can be accessed from script file using the command:

AFOCUS,p

Where p is the parameter set to a number between 0 to 9.

7.3 Auto Alignment

ProbeMagic™ combines various Machine Vision functions to implement the Auto Alignment feature. In the setup process, you will be required to specify two alignment mark templates (left and right) and two search areas. Auto alignment is set from the **Parameter Property** sheet, **Alignment** page.



When the **Auto Align** check box is checked, the feature will be enabled. The **Pass Score** will set the acceptable criteria for the alignment mark recognition. The **Left Area** button starts the process to define the search area for the left alignment mark. The **Show Tmp.s** button displays the left and right alignment mark template images. The **Right Area** button defines the search area for the right alignment mark.

The **Acceptance** and the **Max. Trial** parameters are only applicable when the motorized theta option is being used. Unlike software theta, motorized theta physically rotates the wafer/panel to achieve alignment. This means that the alignment process may

have to be repeated several times to achieve good alignment results. The **acceptance** parameter sets the acceptance level in terms of the degree of the last mechanical theta adjustment. The figure obtained determines whether or not another alignment process is necessary. For example, given that the acceptance parameter is set to 0.1 degree, the machine proceeds to find the left and right alignment marks and then calculate the angular offset of the wafer. Assuming that the found angular offset of the wafer is 0.5 degree, the machine will rotate the motorized theta by 0.5 degree. But, instead of declaring that the alignment process is complete, the machine will repeat the alignment process again because 0.5 degree is larger than the acceptable parameter of 0.1 degree. The **Max Trial** parameter sets the maximum number of times that the machine is allowed to repeat the alignment process to bring the adjustment within acceptance before declaring alignment unsuccessful with an error message.

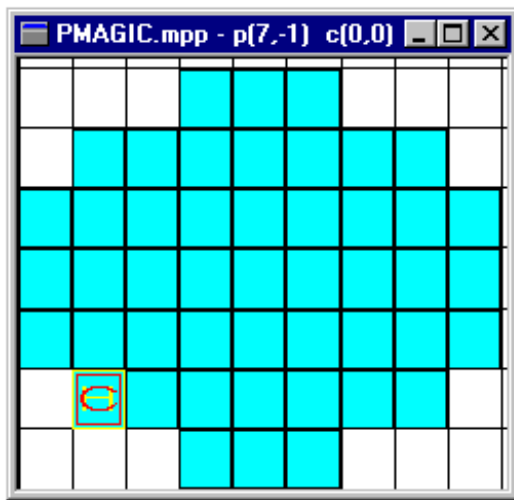
Note: When **Auto Align** is enabled, the **Learn** button will start the alignment marks learning process (template training).

8.0 User Programs For Automation

In addition to all of the interactive features that speed up setups and empower analytical operations, ProbeMagic™ provides three program modes for users to implement automated processes.

8.1 Map Programs

Map programs are suitable for probe scenarios that involve a great deal of repetitive action on a single wafer/panel. For example, if your wafer/panel consists of dice (or clusters) and you need to do the same test on all or some of the dice, or you want to probe all of the row or column lines on a panel to check for open/short, the map program is perfect. These two scenarios will be used to illustrate the function of the Map Program.




In our first example, the product is an 8 inches semiconductor wafer. The requirement is to probe the circuitry of each die. Each die is 22mm by 28mm.


First we set the Die size of the map view to be 22mm in X and 28mm in Y. To do so, we select **Edit Map Data as Text** from the popup menu. An editor window will open up and we can type in the desired die sizes, as well as make changes to other parameters listed. Remember that all dimensional parameters need to be entered in microns.

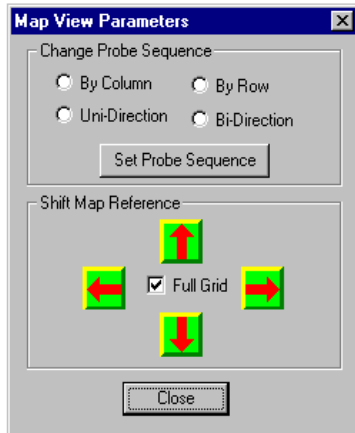
```
[Map0]  
BoardSi
```

```
zeX=200000  
BoardSizeY=200000  
DieSizeX=22000  
DieSizeY=28000
```

....

In this example, a probe card is made to contact the appropriate pads on the die. To select the die to probe, you need to first select from the popup Menu, **Select Die to Probe** to enter the probe dice selecting mode. In this mode, the mouse cursor changes shape to  **P**. To select, point the mouse to the die you want to probe and click the left mouse button. To select a range of dice, you can depress the left mouse button and then drag it across the dice you want to select. Upon release of the mouse button, the range of dice will be selected. The selected probe die will be marked with a blue rectangle. After selecting the die, select **Select Die to Probe**, from popup menu again, to toggle back to normal mode. If you mistakenly selected a die or some dice that was not supposed to be selected, you can select **Un-select Die** from the menu. This mode is

marked by the mouse cursor . In this mode, whichever die you had picked will be de-selected. After setting up the probes, select **Run Map Program** from the popup menu. The probing will start and the stage will move the substrate so that the probe touches down on the entire selected die and conduct the measurement.

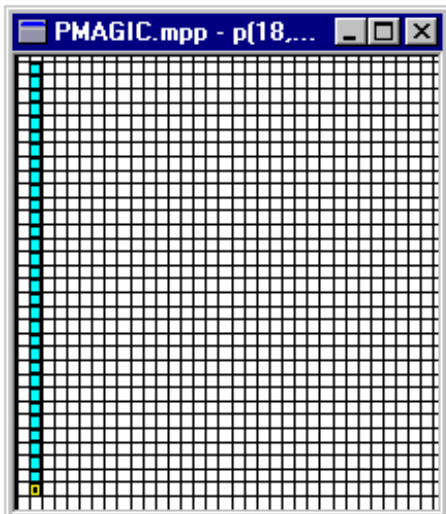


Note: You may want to change the probing sequence by selecting **Edit Parameter** from the pop up menu to launch the parameter dialog.

The measurement instruments are controlled using a script file. In this example, a HP4145B is used to conduct semiconductor parametric measurements. The Metrics® software is integrated to perform the instrument setup and data collection. We will talk about the associated script file later in this section.

You will also need to conduct a second test on the same panel. This test involves very sensitive measurements where a pair of triaxial probes is used. (Probe cards give off too much leakage and cannot be used in very sensitive measurements.) The probes need to touch two pairs of pads on the die. This is called within die probe movement. To achieve this, we can either modify the script routine to move the probe or use a Test Site program.

In this particular example, to achieve the same probing, it is easier to just add a move relative (MR) command in the script routine as opposed to using a Test Site program. ProbeMagic™ gives you the choice to pick either technique.



In our second example given above, the application desired is to probe all the row lines on an LCD panel. All lines have a 100 um separation from each other. A pair of membrane probe heads is used to step through all of the lines. The probe head contacts 30 lines at a time. (To compensate for the XY stage's flatness variation, the probe head assembly for holding the membrane probe array is spring-loaded. Unlike regular probe cards, a membrane probe array cannot have edge sensor for contact sensing).

First you set the X grid size to be 1500 um. Then select the grid to probe. (They are marked as green rectangles on the picture on the left.) After setting up the probe heads and measurement instruments, **Run Map Program** needs to be selected from the popup menu. The software will do all of the stepping and test all lines. The details of instrumentation

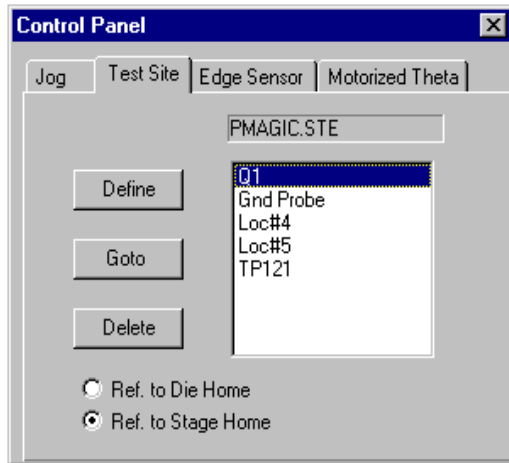
for this setup will not be discussed here.

A Map Program can be associated with a Test Site program or Script program to suit more complex test scenarios. After the association, the Test Site program or the script program

behaves like a subroutine of the Map Program. To assign an associated file to the current map program, select **Edit Data as text** from the map view popup menu and then add the file name after the word, **AssoFile**. For example:

AssoFile= MyFile.scr

8.2 Test Site Programs



A test site program gives you more freedom in terms of probing locations. It is suitable for places where panels are not composed of many repetitive patterns. For example, if you have built a LCD panel on which there are 9 special test structures, and you want to create a program to test the circuits in all 9 of the structures to determine uniformity, a Test Site Program would be most appropriate.

A test site program is created by learning the position of each test site. To do so, you must move the stage to the desired position and press the **Define** button on the **Test Sites** page of the Control Panel. The software will then prompt the user for the name of the test site. A valid test site name is an ASCII string that is no longer than 10 characters and contains no spaces between characters. A Test Site program is stored on disk as an ASCII text file and the parameters of a Test Site program can be changed by editing the text file.

The test site program page has its own local pop up menu. To display the menu, simply point the mouse on the page and depress the right mouse button. The menu contains the following selections:

- Run** - Runs the Test Site Program
- Edit Data** - Uses an ASCII text editor to edit the test site data.
- Load Test Sites** - Loads the Test Site program from file.
- Save Test Sites** - Saves the current test sites in a disk file.

To edit the parameters of the currently loaded Test Site program, you can select **Edit Data** from the pop up menu. The Editor will automatically launch with the proper file loaded. When the editor window is closed, the changed parameters will take effect. Here is a sample of a Test Site program:

```

[TestSiteProgram]
Dwell_Time          = 1000
Associated_Script_File = NONE
TestSitesLocalToDie = 0

[TestSites]
Q1          0.000    0.000
TFT#23     10000.000 10000.000
GND        0.000    20000.000
TP1        30000.000 20000.000
TP12       20000.000 10000.000
Q6         20000.000 10000.000
VOUT       20000.000 10000.000
[End]

```

A Test Site program can be associated with a script program to perform more complex operations such as instrument control software interface and having multiple pads probe within a test site. After the association, the script program becomes a subroutine of the Test Site program.

Going back to the previous example, if you need to move the probes within the test structure to do three different tests on six different pads, then an associated script program will need to be used. An associated script program must contain at least one of the following three subroutines for it to be able to associate with a Test Site program. The subroutines are INIT, TEST, and END. The INIT subroutine is called prior to visiting any test site. In this subroutine, you should do all initialization that is needed before the test. This subroutine will be called only once during the execution of a Test Site program. The TEST subroutine is called every time the system arrives at a test site. You should put the test related script command (e.g. command to invoke the Test Instrument, etc.) in this subroutine. The END subroutine is called after all of the test sites have been visited and is the location for any clear up commands. To assign an associated script file to the current test site program, select the **Edit Data** menu, and then edit the line, Associated_Script_File. For example:

```
Associated_Script_File = MyFile.scr
```

8.3 Script Programs

A script program is composed of a list of commands and is the most flexible program mode. It allows you to use all of the functions of the LCD probe stations. All of the example test scenarios discussed in the Map program and Test Site program sections can be completed using a Script Program. Although script programs can be used independently, they are more frequently used as an associated file for a Map or Test Site Program, because of the difficulty in creating a script program.

In a script program, any of the interactive and batch commands can be used. With the addition of program control features, the script program becomes a powerful application specific language.

Below is an example of a script program. This and many others (DEMO1.SCR to DEMO5.SCR) are presented in the ProbeMagic™ software package and are also listed in the appendix. In order to learn more about script programming, it would be best to modify the given examples. To edit a ProbeMagic™ script program, select from the menu **File, New Script**. The ProbeMagic™ Editor will appear.

In this first example, the stage is commanded to move in a square pattern.

```
; EX1.SCR
;
; A simple script program. It draws a rectangle.
;
SPEED,XY,50      ; Sets speed to 50 mil per sec
ZERO             ; zero position counter
MR,60,0          ; Moves relative
MR,0,30
MR,-60,0
MR,0,-30
```

Variables

Up to 30 user definable variables can be used in ProbeMagic™ script programs. All user variable names have to be declared using VAR command before use. Variables can be used as command parameters, such as with the following commands CALL, T, LOOP, N. Variables can also be used in math expressions, as well as be printed or saved to files.

Labels

Up to 20 alphanumeric labels can be used in a script program. All labels must start with a ':'. Only the first five characters in a label string are stored. To be unique, a label must differ in the first five characters from all other labels used in the program. Labels are case insensitive, meaning that there is no difference between :HI and :hi, as far as the script program is concerned. The following are examples of valid labels.

Example:

```
:L1
:LABEL
:GOOD
```

:Fail

Labels are used with the GOTO command to jump program execution to another portion.

Math expressions

It is also possible to evaluate math expressions in script programs using the LET command. The operations +,-,*,/,<,^,(, and),OR, AND, NOT, <,> ,= are allowed. A typical expression, LET A=B*(16+C) assigns to the variable A the value of the expression that follows. Operations include,

OR	Logical OR	(Lowest priority)
AND	Logical AND	
NOT	Logical NOT	
<,> ,=	Relational	
+	Addition	
-	Subtraction	
*	Multiplication	
/	Division	
^	Raise to a power	
()	Order of operation	(Highest priority)

Operations are calculated in an algebraic fashion. First, all values in parentheses are calculated, and then all numbers are raised to powers. Next are multiplication and division operations and finally all addition and subtraction operations are carried out. After all mathematical operations are completed, relational operations are performed and lastly, logical operations are done.

Note: Math expressions can also be used as command parameters, e.g. MV,A*3,(B-5).

Data file access

Using the **READ** command, it is possible to read data from a standard ASCII file. Data is stored in a comma delimited format. Values are separated by a comma or an end of line in a normal ASCII file. Comments can be inserted using a semi-colon (;). All text after the semi-colon is treated as comment. Data files are opened by using the **OPEN** command and are closed using the **CLOSE** command.

Example: Data file format

```
312,500 ; starting location
101 ; miscellaneous data
200,210 ; another location
200,220 ; this and other comments above are ignored
```

Decision making

The IF command can execute one of two statements, depending on the result of a logical operation in a script program.

Looping

The LOOP command executes a series of commands repeatedly. The number of executions can be fixed by setting a number to a variable.

Logical operations

Using the LET command, you can evaluate logical expressions in ProbeMagic™ script files. The operations +,-,*,/,^,(, and),OR, AND, NOT, <,>,,= are allowed.

OR	Logical OR
AND	Logical AND
NOT	Logical NOT
<,>,,=	Relational greater than, less than, equal to

The user may compare results of statements and data and perform conditional branching using the IF command.

Subroutines

ProbeMagic™ script files can use subroutines to perform repetitive operations. Subroutines must be placed at the beginning of a program. Subroutines start with the declaration SUB,name. The desired operations are then included and the subroutine is ended with a RET instruction.

8.31 An Associated Script Program

When a Script program is used as an associated file for a Map or Test Site program, it must contain at least one of the following three subroutine definitions: INIT, TEST, END.

INIT is called before the probing begins, where in it, you should put any initialization commands. The TEST subroutine is called every time a die or a test site is reached. In this, users should put all the test related commands such as commands to invoke the test instruments. The

END subroutine is called after all the die or test sites have been visited. Here is a very simple associated script program:

```
SUB,TEST
ZMV,0           ;touches down the probes
DELAY,1000     ;waits 1 second
ZMV,0.5        ;lifts the probe
RET

BEGIN
END
```

Here is another example of a script file that is associated with a Test Site program. In this test, the Metrics software that controls a HP4145 to perform parameter analysis of MOSFETs is used. ProbeMagic™ communicates with the Metrics® software via a Windows® DDE link.

```
SUB,INIT
DDEINIT,METRICS,COMMAND ;establishes communication with Metrics
MV,0,0                 ;moves stage to home position
CSENSOR,ON             ;enables contact sensor
ZMV,0.1                ;lifts probes
RET

SUB,TEST
ZMV,0                 ;touches down probes
DDEEXEC,[MEASURE]     ;starts measurement
DELAY,2000            ;waits for measurement to complete
ZMV,0.1               ;lifts probe
RET

SUB,END
DDEEND                ;ends communication with Metrics
RET

BEGIN
END
```

9.0 System Integration

There are many scenarios in which ProbeMagic™ has to work with other equipment or instruments to accomplish a task. ProbeMagic™ was designed with system integration in mind. There are several ways that ProbeMagic™ can interface with other systems.

ProbeMagic™ supports MS Windows® OLE Automation that allows Windows® programs to communicate with each other on the same computer. There are also remote interfaces that allow a remotely located host computer to control ProbeMagic™. All interfaces use a generalized command set to communicate with ProbeMagic™. In this command set, there are four groups of commands.

1. Interactive Commands

All ProbeMagic™ commands that are typed into the Command Window can be sent over via an interface. These commands are **MV, MR, SPEED,RUN**, etc.

2. Status Query Commands

- @GP axis - gets current position (example response: 123.23)
- @GS axis - gets speed settings (example response: 40.00)
- @GD - gets current die location (example response: c(12,2))
- @GT - gets current test site name (example response: LOC#1)

3. Host Interface Commands

- SOT - Start of test, ProbeMagic™ sends to host.
- EOT - End of test, host sends to ProbeMagic™.

9.1 MS Windows® DDE and OLE Automation

Microsoft Windows® is a multitasking environment in which multiple programs are run concurrently. The two primary ways of inter-process communication are DDE (dynamic data exchange) and OLE (Object Linking and Embedding). OLE has recently been renamed Active X.

DDE Client

The DDE client initiates a DDE conversation. The client determines when and what the conversation is about, and the server merely responds to a client's request. In other words, a client "controls" a server. ProbeMagic™ implements the client functions by the following five commands:

- DDEINIT** - Initiates a DDE Conversation
- DDEEND** - Ends a DDE Conversation
- DDEEXEC** - Sends a command to the server
- DDEPOKE** - Changes a data item in the server
- DDEREQS** - Reads a data item from a server

For example, in the situation that a user wants to use ProbeMagic™ to control the probe station and the instruments, but desires to use a Windows® spreadsheet program (e.g. Microsoft Excel) for data analysis and charting, this can be done by using a script program similar to DEMODDE.SCR. This is included in the ProbeMagic™ software package.

Today's microelectronics testing usually requires complex instrument controls. The degree of complexity can be more than ProbeMagic's™ built in script commands can handle. In those cases, specialized instrument control programs (e.g Metrics & ICS) can be used as script files using DDE client commands under ProbeMagic™.

DDE Server

In other scenarios, it is better to have the Instrument Control Software control ProbeMagic™. This way, ProbeMagic™ and the probe station become one of the instruments. The Instrument Control Software commands ProbeMagic™ to move the probes to a test site and touch down the probes. The software then commences making measurements, then collects and analyzes data. After the testing is finished, it commands ProbeMagic™ to move to the next test site. The cycle repeats until all the test sites have been tested.

To have ProbeMagic™ controlled by other programs via DDE, ProbeMagic™ must be a DDE server. The client can then send any ProbeMagic™ command to the ProbeMagic™ software using DDE EXECUTE. This includes the RUN command that can start a script file. The client can also query the status of ProbeMagic™ by requesting the value of DDE items in ProbeMagic™. There are nine supported built-in items:

- XPOS - X position of the current stage
- YPOS - Y position of the current stage
- ZPOS - Z position of the current stage
- SPEED - Current stage speed.
- ACCEL - Current stage acceleration.
- TEST - The test status either: EOT (end-of-test) or SOT (start-of-test)
- EOW - End of Wafer Status: can be 0 or 1
- DIELOC - Current die location on the map view.
- TSITE - Current test site name
- FSTS - File Status: can be "Stopped", "Paused" or "Running"

The DDE server name and topic name for the ProbeMagic™ software are respectively, PMAGIC and COMMAND. This implementation of DDE allows ProbeMagic™ to communicate with any Windows® application capable of DDE. Section 9.3 discusses such an interface to HP VEE.

OLE Automation Server

ProbeMagic™ is also an OLE Automation server with the name **PMAGIC.SERVER**. OLE automation is currently the most efficient way to communicate with other OLE capable Windows® software, namely Visual Basic or Visual Basic for Applications, which are both embedded in Microsoft® Excel and Microsoft® Access.

The following methods are available:

ExecCmd(str)

Executes a ProbeMagic™ Script command.

e.g. PM=GetObject("PMAGIC.SERVER)
 PM.ExecCmd("RUN,Test.Scr")

ErrorMsg(str)

Sends an Error Message for ProbeMagic™ to display.

MoveAbs(x,y)

Moves absolutely to location (x,y) where the unit of measurement is in microns.

e.g. PM=GetObject("PMAGIC.SERVER)
 PM.MoveAbs(12000,34000)

MoveRel(dx,dy)

Moves relatively in dx and dy microns.

e.g. PM=GetObject("PMAGIC.SERVER)
 PM.MoveRel(2000,4000)

The following properties are available:

XPOS	- X Axis position in microns
YPOS	- Y Axis position in microns
XYSpeed	- XY speed in microns
MotionDone	- is TRUE if the motion has been completed
SOT	- is TRUE if Start of Test is signaled
EOW	- is TRUE if End of Wafer is reached
TESTSITE	- Current Test Site Name
DIELOC	- Current Die Location.
FSTS	- File Status (can be "Stopped", "Paused" or "Running")

The following is a Visual Basic for Application code segment that illustrates the interface with Excel Spreadsheet using the OLE Automation. The spreadsheet (**OLETest.xls**) can be found in the **Examples** sub-directory.

In this example, we control the GPIB instrument directly from within the Visual Basic code with functions such as **ibfind**, **ibwrt** and **ibrd**. These functions are implemented by National Instruments® in **GPIB-32.DLL**. Two files, **NIGLOBAL.BAS** and **VBIB32.BAS**, have

to be imported into the spreadsheet module. These two files together, along with the **GPIB-32.DLL**, are provided in the software package accompanying any GPIB boards build by National Instruments®.

```
Dim PMagic As Object
Dim HP3478A As Integer
```

```
'Get reading from the instrument
Function DoTest() As Integer
    Dim rstr As String
```

```
    ibwrt HP3478A, "F1"    'Sets Function Measure DC Voltage
    rstr = "                "
    ibrd HP3478A, rstr    'Gets Reading
```

```
    ' r = Int(Rnd * 3) + 1
    r = Val(rstr)
    ActiveCell.FormulaR1C1 = r
    ActiveCell.Offset(1, 0).Activate
    DoTest = r
End Function
```

```
'This procedure is call periodically
```

```
Sub TimerTick()
    Dim rstr As String
    If PMagic.SOT = True Then 'Checks Start of Test Signal
        result = DoTest()    'Performs the testing
        rstr = "EOT," + str(result) 'Sends End of test with bin info
        PMagic.ExecCmd (rstr)
    If PMagic.EOW = False Then 'Checks for End of Wafer Status
        Application.OnTime Now + TimeValue("00:00:01"),"ThisWorkbook.TimerTick"
    End If
    Else
        Application.OnTime Now + TimeValue("00:00:01"),"ThisWorkbook.TimerTick"
    End If
End Sub
```

```
' This is the main procedure to start the interface with the ProbeMagic™
```

```
Sub PMagicInterfaceMain()
    ibfind "Dev1", HP3478A 'Connect to DMM on the GPIB Bus
    Worksheets("Sheet1").Range("A1").Activate
    Set PMagic = GetObject("PMAGIC.SERVER") ' Connect to ProbeMagic™ OLE Server
    PMagic.ExecCmd ("Run,WAFER") 'Start Execute the wafer map
    Application.OnTime Now + TimeValue("00:00:01"), "ThisWorkbook.TimerTick"
End Sub
```

10.0 Saving and Restoring Setups

With ProbeMagic™, a test setup can be saved to disk under a name. This feature allows for easy switching between tests and multiple individuals to use the system without affecting each other. To save a current test setup, select from the menu bar, **File, Save**, and enter the name of the setup with the extension .psu. Then click OK. To restore a test setup, select from the menu bar **File, Open**. Then pick the name of the setup file, with extension .psu. Finally, click OK.

The setup information includes the home position, alignment marks definitions, jog step sizes, jog speed, move speed and acceleration, any map view definition or program, any test site definition or program, function key definition, button bar user defined buttons, any option or user written DLLs information, and any option's setup parameters (e.g. laser parameters, vision system parameters), etc.

Since different categories of information are stored, setups are stored in multiple files with the same name, but with different extensions. For example, if you want to save all of the information related to the testing of panel P101 under the name P101, the following files may be created automatically by the computer:


- P101.PSU** This file has the same format as PMAGIC.INI. It saves the home position, alignment marks definition, and information about what options or DLLs need to be loaded.
- P101.MPP** This file is created only if there is a map view program in the current setup. This file saves the map view program.
- P101.STE** This file is created only if there is a test site definition in the current setup. This file saves the test site program.
- P101.LSU** This file is created only if the laser option is installed. This file saves all the laser settings

All the files in a setup can also be individually saved and restored. This allows for easy creation of new setups from existing setups. In the case where the user wants to create a setup for P102, and the user knows that the P102 setup differs from P101 setup only in the test site definitions, s/he can load the setup for P101, and re-define the test sites. Then s/he will only need to save the setup under the file name P102.

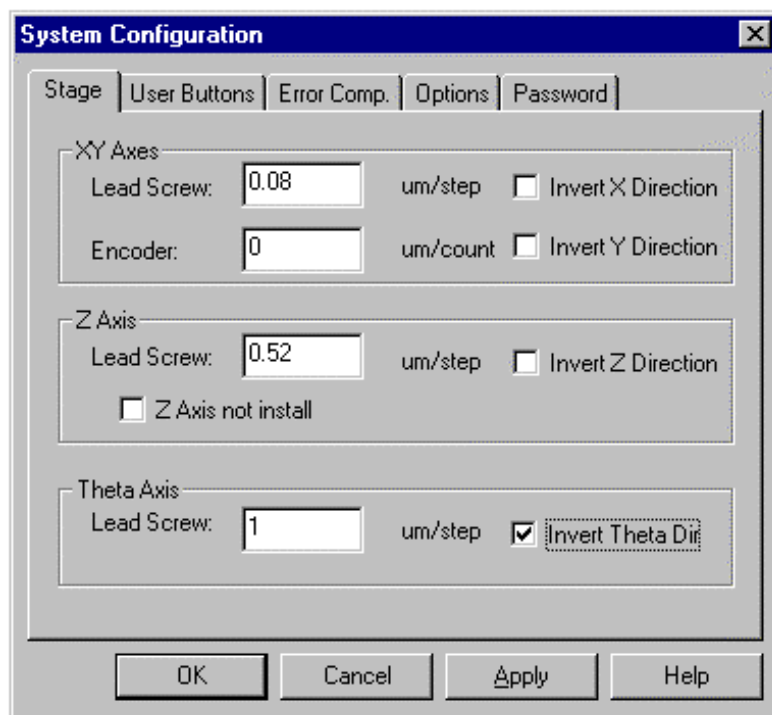
In another example, if the user knows that the P103 setup has the same test site definition as P101, but all of its remaining information is the same as P102, then, the user can load the setup P102. Then, the user can load the test site file, P101.STE, and save the setup under the name, P103.

11.0 Configuration and Startup

Most of the ProbeMagic™ settings, such as speed, scale, DLL loaded, button bar buttons, etc., are automatically saved upon program exit. These settings will be restored when the program is restarted. However, some other less frequently changed settings, such as lead screw pitch, encoder resolution, error compensation factors, axis directions, etc. can only be changed through the configuration process.

To re-configure, type **CONFIG** in the **Command** window, (if the window is not displayed click on the  button). You will then be prompted for a password. The default password is PMAGIC, which can be changed in the configuration process. After getting through the check, the password, following the property sheet, appears. There are multiple pages in this

property sheet. Each page corresponds to one category of the configuration information.



To set the XY stage settings, press the Stage tab.

The page box contains all the configurable motion settings:

1. The lead screw pitch is specified in microns per motor step. (The setting is a product of motor resolution and lead screw pitch.) For example:

For a 2 mm per revolution lead screw, 1.8 deg motor, and a 125 micro step per step micro stepping motor driver, the value is:

$$\frac{2 * 1000}{125 * 200} = 0.08$$

2. The encoder resolution specifies the number of microns per encoder count. Enter 0 if the encoder is not used, meaning, an open loop system is being used.

3. The axis direction allows for you to reverse the travel direction of an axis. This is sometimes necessary to match the substrate coordinate system.

For more detailed configurations, you can edit the file, **PMAGIC.INI**. (The PMAGIC.INI file is in the directory where the PMAGIC.EXE is located, usually in

C:\PMAGIC\BIN.) You should change the **PMAGIC.INI** file using ProbeMagic's™ build-in editor. To load the file, select from the menu **File, Open**, then type in **pmagic.ini**, and press the OK button.

If you use other text editors, such as NotePad, to edit pmagic.ini, you should only edit the file when ProbeMagic™ is not running. Since ProbeMagic™ always overwrites the PMAGIC.INI based on the current settings upon program exit, changing PMAGIC.INI while ProbeMagic™ is running will cause the changes to be overwritten.

In the configuration process, you can setup a startup file. A startup file is a ProbeMagic™ script program that will automatically execute when the ProbeMagic™ software starts. It is similar to the AUTOEXEC.BAT file for DOS. This can be used in situations where you want to initialize several GPIB instruments every time ProbeMagic™ starts. The default startup file is named "Startup.scr".

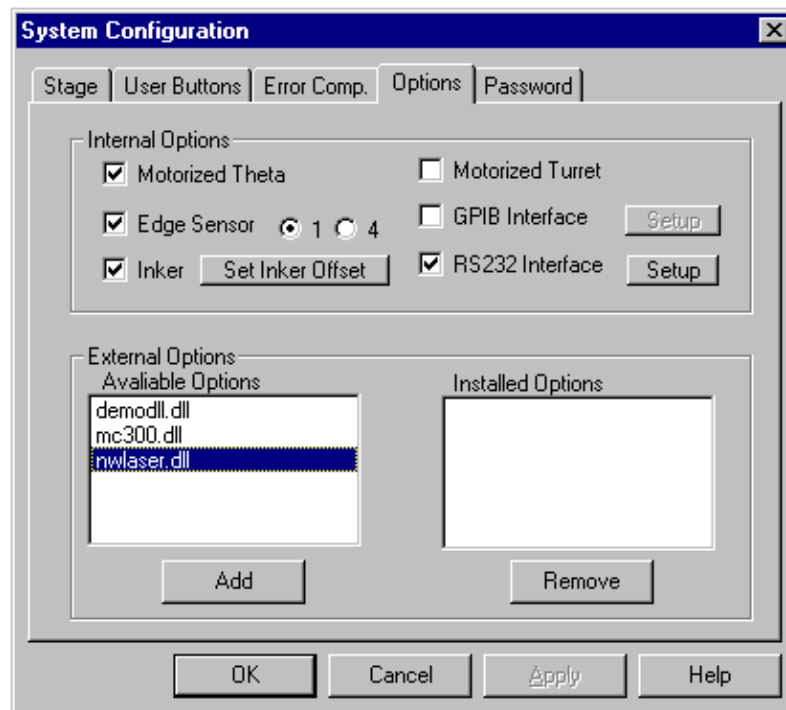
12.0 Open Architecture

In today's high tech industries, customer requirements change very quickly. A piece of software, no matter how well thought out during the initial design process, is bound to become inadequate in a few years. In order to give ProbeMagic™ a projected live span of over 5 years, it has been given an open architecture. The architecture allows ProbeMagic™ to expand its functionality by linking in option DLLs (dynamic link libraries), which makes the external option DLLs an extension of the core software.

This modular approach makes ProbeMagic™ more customizable to better satisfy our customers. We can implement customer specific functions through option DLLs and keep the core software for all of the customers the same.

An option DLL can add new commands to ProbeMagic's™ command list, where once a command is added, it can be used as any other Interactive command. It can be used in a script file, typed in from the Command Windows, sent over from a remote host computer or sent from another program using OLE. An option DLL can also add buttons into ProbeMagic's™ Button Bar. The button can then be used to access functions in the option DLLs.

To load in a new external option DLL, launch the **configuration** property sheet and select the **Options** page.



The information regarding options is saved in the PMAGIC.INI file and the ProbeMagic™ Setup files (*.psu). When ProbeMagic™ starts, it will load the appropriate option DLLs based on the information saved in the PMAGIC.INI file. A similar process takes place

when the system setups are changed. By opening a .psu file using menu selection **File, Open**, option DLLs will be loaded or unloaded based on information stored in the setup files.

Nearly all of ProbeMagic's™ add-on options are implemented through DLLs. For example, the laser option is implemented through the file, nwlaser.dll. DLLs communicate with the core software using the same interface as any other user written DLLs. Theoretically, a ProbeMagic™ DLL can be written using any program language or compiler. However, the C++ language is recommended as this is the language that ProbeMagic™ is written in.

13.0 Motion Emulation Mode

The ProbeMagic™ software has an emulation mode. In this mode, ProbeMagic™ emulates stage motion, DDE interface, DLL interface and the remote interfaces, RS232 and GPIB. This feature is requested by many of our customers who want to get a head start on using the probe station while waiting for the delivery of the system. (Normally, it takes KORIMA 6 to 12 weeks to deliver an LCD probe station or laser repair system.) During this period, users with the ProbeMagic™ software package can run the software in emulation mode on their own computers. This will allow them to become familiar with the software and the operations of the probe station.

ProbeMagic™ emulates stage motion according to the speed and acceleration settings inputted by the user. This allows you to do throughput analysis. ProbeMagic™ also emulates the tester software (Instrument control software) interface, and GPIB interface (require a GPIB board in the computer), allowing one to literally develop the entire instrument control software or remote host control program using the emulation software.

Even in emulation mode, you can develop ProbeMagic™ map, test site and script programs. All of the programs developed in the emulation mode will run without modification also on the actual system. In addition, by running the emulation software, the users will gain important insight to the operation of the actual probe station. This allows prospective users to understand the functionality of the probe station before it is delivered, making it possible for one to request desirable additional software features to arrive concurrently with the probe station.

What is emulated?

Most of the ProbeMagic™ features are emulated. They include: Jog motion, joystick motion, command motion, map view motion, test site motion, map program, test site program, script programs, associated script programs, panel alignment command, DDE server and client functions, the EOT and SOT commands, external DLLs loading, button bar button definition and edit, save and load setups, labview interface, Metrics interface, ICS interface, HP VEE interface, remote interface RS232, GPIB (if GPIB hardware is present), etc.

What is not emulated?

Some of the highly interactive features, such as live video, video navigator and on-screen dimension measurement, are not emulated. However, in emulation mode, the software will display a stored bitmap file (emulate.bmp) on the live video window. We hope this will give the user a feeling of the live video display. The emulate.bmp is a 256 color bitmap, thus if the VGA mode only supplies 16 colors, the display will be highly degraded.

Note: The live video board will always display live video in true color (16 million colors) regardless on the VGA mode of the computer.

14.0 Error Compensation

14.1 Linear Error Correction

Positioning accuracy is very important for probing applications. For large area LCD panel probing, accuracy becomes a very demanding specification. To enhance positioning accuracy, ProbeMagic™ compensates for mechanical errors of the positioning system. During the configuration process, one can enter correction factors for error compensation. ProbeMagic™ compensates two kinds of errors:

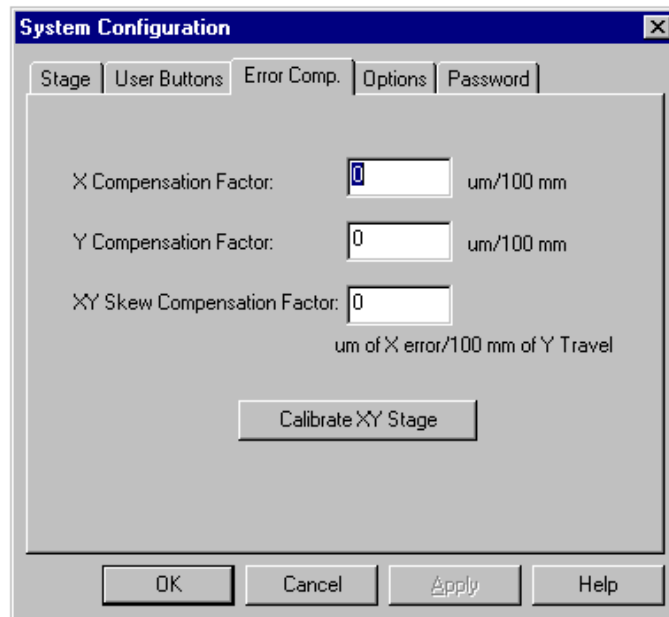
1. Linear over or under travel

Correction factors are in microns per 100 mm. For example, if an axis over travels 5 μm over 100 mm, -5 should be put in as the correction factor of that axis.

2. Orthogonality error

On some stages, the X & Y axes are not perfectly perpendicular. Traveling over one axis results in a skew shift in the other axis. This is the so-called skew error. Skew error correction factors are also in microns per 100 mm. This would mean that, if for 100 mm of Y travel, the X shift to the right 3 μm , -3 should be entered for the correction factor.

The Linear error compensation factors are set up using the CALISTAGE command or via the **System Configuration** property page **Error Comp.** Page:



14.2 Pattern Recognition

Pattern recognition is a very handy tool for error compensation during wafer probing.

15.0 Trouble Shooting

A lot of work has gone into testing the ProbeMagic™ software, but with its complexity, there is always the chance that software bugs remain unfound. KORIMA, Inc. appreciates users input, thus if you think you have discovered a software bug or have some suggestions, please feel free to fax or call KORIMA.

Software Crashes

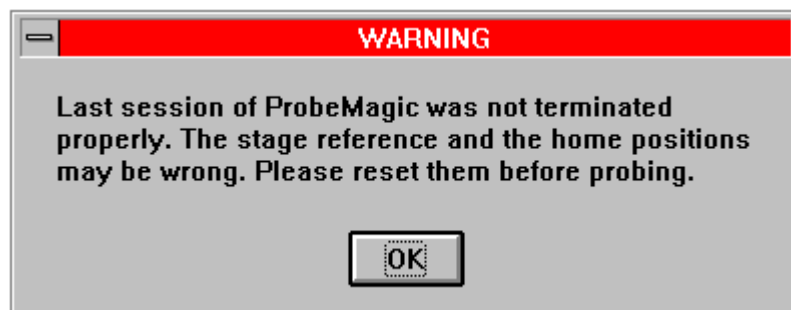
The most serious type of problem is a software “crash”. In Windows®, a “crash” is signaled by displaying a white box covering most of the screen area with message such as “General Protection Fault” or “Stack Overflow,” etc. The box usually has two buttons, **Close** and **Ignore**. In most cases, you can only close the offending application. A crash can be caused by ProbeMagic™ itself or by other concurrently running applications. We have noticed that an incorrectly configured instrument control software, such as LabVIEW or HP VEE, may exhaust system resources and cause ProbeMagic™ to crash or lockup. We are still working to identify and resolve such situations.

If you cannot re-start ProbeMagic™ after a crash, you will need to exit and re-start Windows® 95 and then try to start ProbeMagic™ again.

If ProbeMagic™ crashes during startup, one of its configuration files is probably corrupted. To resolve the situation, delete the file **pmagic.ini** in the ..\BIN sub directory (e.g. C:\PMAGIC\BIN).

ProbeMagic™ will start up again using default settings. To restore your configuration, load one of your previously saved setup file (*.psu). To do so, select from the ProbeMagic™ menu **File, Open**. If you haven't saved any setups, load the file, **backup.psu**, found in the BIN sub-directory.

After ProbeMagic™ starts, it will try to find out if the previous session was ended improperly, such as in the case of a crash. If it detects so, it will remind you to set the stage reference again. Any abnormal termination of the software will result in the losing of the stage reference position. The software will display the following message to remind you to set the stage reference again before proceeding:



COM3 Serial Port Problems

With KORIMA Inc.'s LCD probe stations or laser repair systems, the COM3 port may be used to control the laser cutter or interface with other devices. The IRQ of COM3 is set to IRQ9 (since COM1 at IRQ4 is used for the mouse, COM2 at IRQ3 is used to control the autofocus and motorized turret circuit). Since IRQ9 is not the default IRQ for COM3, if you later re-install MS Windows® on the computer, you will need to configure the IRQ for COM3 back to IRQ9 for proper function.

To configure the IRQ for COM3:

1. From Windows® 95 **Start Menu**, click the **Settings, Control Panel**.



2. Double click on the System icon **System** to start **System Properties**.
3. Select the **Device Manager** tab.
4. Click on the + sign before the **Line Ports (LPT & COM)**
5. Double click on the **Communication Port COM3** to display the property sheet for COM3.
6. Select the **Resources** tab.
7. Change **Interrupt Request 9**.

Problems with the Mouse

If you use a bus mouse on the computer running ProbeMagic™, you may experience mouse problems. This is because ProbeMagic™ uses IRQ5 to communicate with the motion control board. To resolve the problem, set the bus mouse IRQ to 2.

Network Problems

If you install a network card onto the KORIMA computer and experience problems, there may be an IRQ or I/O address conflict. KORIMA computers have the following IRQ and I/O address setup:

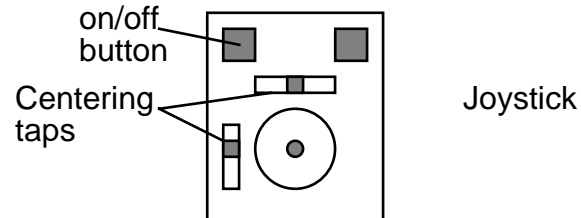
IRQ3	- COM2 (autofocus and turret)	I/O address: 3F8 hex
IRQ4	- COM1 (mouse)	I/O address: 2F8 hex
IRQ5	- OMS motion control board.	I/O address: 300hex - 308 hex
IRQ7	- GPIB board (if installed)	I/O address: 2B8 hex
IRQ9	- COM3 (laser cutter or accessory device)	I/O address: 3E8 hex

The network adapter usually uses IRQ 11. In some computers, when IRQ9 is used, it will block IRQ10 to IRQ15. This will cause the network adapter to stop functioning. To resolve this problem, the COM3 IRQ needs a high unused IRQ number such as 10 or 12. This sometimes means changing the board because not all serial boards support IRQs higher than IRQ9.

Joystick Problems

If you are using a standard joystick and experiencing problems, such as only being able to move the joystick in one direction on one axis, you may need to re-center the joystick.

A joystick can be re-centered as follows:



Note: Steps must be undertaken while ProbeMagic™ is running.

1. Exit joystick mode by pressing the on/off button on the joystick. You should see the message, “joystick off” in the status bar.
2. Adjust the centering tap of the troubled axis.
3. Enter joystick mode once again by pressing the on/off button again. You should see the message, “joystick on” in the status bar.
4. Try out joystick functions. If the problem still persists, repeat this procedure and try a different position for the tap.

16.0 ProbeMagic™ Commands

There are two main types of commands.

1. **Interactive**
2. **Batch**

(Commands marked preliminary are scheduled to be completed in future versions of ProbeMagic™. They may not function correctly if used.)

16.1 Interactive Commands

The following commands can be executed interactively by entering the Command window or by being sent over a remote interface (GPIB or RS232) or OLE Automation link. They can also be used in a script file.

Here is a list of all the interactive commands in alphabetical order:

ABORT

Stops all stage motion immediately.

If there is a laser option, the laser is turned off.

Pressing the **STOP** button on the Jog page in Control Panel, and the **STOP** button on the ToolBar will have the same effect.

ACCEL,axis,x

Sets the stage acceleration in user units/sec/sec.

Where: axis - X,Y,Z,XY,ALL

x - acceleration in user units/sec/sec

Example: ACCEL,XY,500

Sets the XY axis's acceleration to 500 user units per second per second.

You can also change the axis's acceleration by accessing the main menu (**Edit-Parameters**, Stage page)

AFOCUS,p[,mp]

Performs autofocus using parameter.

Where: p – parameter set number (0 to 9)

mp-optional parameter, must pass, if it is set to 1, the system will display a message box to prompt you to focus manually if the autofocus cannot achieve the required pass score.

Example: AFOCUS,0,1

ALIGN[,p1][,p2][,p3]

Aligns the substrate base on alignment mark definitions or set alignment parameters.

Where: p1 is

- M** - uses motorized theta to perform the alignment e.g. ALIGN,M or ALIGN,M,N (alignment without initial move, assuming the left alignment is on the current screen.)
- S** - aligns using software theta; this is the default alignment method i.e. ALIGN is the same as ALIGN,S. However, this parameter is needed if alignment without initial move is desired. ALIGN,S,N
- H** - aligns using the theta alignment knob on the chuck.
- SETLXY** - sets the after align position of the Left align mark, e.g. ALIGN,SETLXY,10,20.
- SETRXY** - sets the after align position of the Right align mark, e.g. ALIGN,SETRXY,200,20

Defining alignment marks is a simple task. If you know the nominal (ideal) locations of the alignment marks, you can enter them into the boxes in the **System Parameters property sheet, Alignment Setup** page, which is launched by selecting from the menu bar, **Edit, Parameters**. If you do not know the position of the mark you can “Learn” them. First put the substrate on the stage, align it using Align-By-Line methods (the SETBASE,M command). Then one can start the alignment marks definition by selecting from the menu bar, **Edit, Parameter**, then **select Alignment Setup** page. Then press the **Learn** button in the page. Two prompt dialogs will display and each will ask you to move to a location on the panel that you want to pick as an alignment mark. In this way, you learn the locations of the left and right alignment marks. Any distinctive feature on the panel can be selected as an alignment mark, e.g. the corners of dice.

In addition to defining the alignment marks using the **Alignment Setup page**, the ALIGN command itself also can be used. This allows changing of the alignment parameters from a script file or from a remote host computer.

The alignment mark definition is save with the setup files and also the PMAGIC.INI file.

ARC,x,y,cx,cy,d (Preliminary)

Moves in an arc.

Where: x,y - the arc start coordinates.
cx,cy - the center coordinates of the arc.
d - angular dimensions of the arc in degrees and arc direction, a positive "d" results in an arc moving in clockwise direction and vice versa.

CALISTAGE

The CALISTAGE command invokes a prompt-guided procedure to determine the three linear error correction factors for the stage. The three error factors are used in position accuracy compensation. Since linear error is the major portion of the total positioning inaccuracy, it is important to complete this procedure correctly. The three error factors are:

1. X scaling error.
2. Y scaling error.
3. XY axes skew error.

Before starting the procedure, you should make sure that the panel on the stage is properly aligned.

The procedure includes four prompt dialogs. All four dialogs are captioned as "Calibrate XY stage accuracy." The first dialog asks you to enter or verify three reference positions on the panel. The nominal or ideal coordinates of the reference points should be entered into the corresponding boxes on this dialog. Any distinguishable feature, including alignment, registration or the corner marks on the panel may be used as a reference point. The first reference point has to be near the bottom left corner of the panel, preferably the (0,0) position or the left alignment mark position. The second reference point has to be near the bottom right corner of the panel. The third reference point has to be near the top left corner of the panel.

After pressing OK on the first prompt dialog, the stage will move to the reference point position on the stage. Don't be alarmed if the stage does not stop near the first reference position. This is because the software assumes that the first reference point is near the left alignment mark. When the stage stops, the second prompt dialog will display. It prompts you to move the stage so that the first reference point lines up with the video cross-hair. You can move the stage by commands (MV, MR), jog dialog arrows or video Point-and-Shoot. After you line up the first reference point, press Ok on the prompt dialog.

The stage will move again to the second reference point. Then another prompt dialog shows up also asking you to move the stage so that the second reference point on the panel lines up with the video cross-hair. Move the stage and line them up and press Ok afterward.

The stage will move to the third reference point on the panel. Please do the same to line up the reference point on the panel with the video cross hair and press Ok on the prompt dialog after you do so.

After visiting the three reference positions, the software will compare the reference points' nominal location and the positions recorded during the procedure to calculate the three error factors.

Users may enter the **System Configuration property sheet** to view the values of the three error factors.

CVEND (Preliminary)
Ends constant velocity mode.

See Also: **CVBEGIN**

CVBEGIN,x,y (Preliminary)
Begins constant velocity mode.

In constant velocity mode, the stage does not stop after each command. Instead, it moves in a continuous fashion to ensure uniform laser spot overlap in applications such as laser machining.

Where: x ,y - positions to start the constant velocity mode.

Example: CVBEGIN,0,0
MV,0,20
ARC,0,20,30,30,360
MV,0,30
CVEND

See Also: **CVEND**

DDEINIT,server,topic

Initializes ProbeMagic to become a DDE client and connects to the specified DDE Server.

Where server - DDE Server name.
topic - DDE topic name

Example: DDEINIT,EXCEL,SHEET1
Connects to Microsoft Excel® spreadsheet SHEET1
DDEPOKE,R1C1,123;change Row 1 Column 1 to 123

See Also: **DDEPOKE, DDEEND, DDEEXEC**

DDEEND

Terminates a DDE conversation that was started by the DDEINIT command.

See Also: **DDEINIT, DDEPOKE, DDEEXEC**

DDEEXEC,cmd

Sends a command string to the DDE server.

Where cmd - command string.

Example: DDEINIT,EXCEL,SHEET1
DDEEXEC,[NEW]
sends [NEW] command to MS Excel®

See Also: **DDEINIT,DDEEND**

DDEPOKE,item,content

Changes a data item in the DDE Server.

Where item - item name.

content- content string

Example: DDEINIT,EXCEL,SHEET1
; Connects to Microsoft Excel® spreadsheet SHEET1
DDEPOKE,R1C1,123;change Row 1 Column 1 to 123

See Also: **DDEREQS, DDEEND, DDEINIT**

DDEREQS,item,content

Reads a data item from the connected DDE server.

Where item - item name.

content- content string

Example: VAR,V1 ;declare variable

DDEINIT,EXCEL,SHEET1
; Connects to Microsoft Excel® spreadsheet SHEET1
DDEREQS,R1C1,V1 ;change Row 1 Column 1 to 123

See Also: **DDEPOKE, DDEEND, DDEINIT**

EOT,[b]

End of Test. When ProbeMagic™ is used in remote mode, the SOT command can be used to signal the host to start testing. After the test is finished, the host should send back EOT to ProbeMagic™.

Where **b** - optional parameter, bin code (1.99)

When a map program is used, bin code corresponds to an ink color on the map.

EXEC,filename

Launches another application from ProbeMagic.

Example: EXEC,pbrush ;launch PaintBrush from ProbeMagic

FMV,p

Moves focus motor absolute.

Where **p** - position to be moved to in um.

Example: FMV,100

See Also: **FMR**

FMR,d

Moves focus motor relative

Where **d** - distance to be moved in um (the unit can be changed)

Example: FMR,10

See Also: **FMV**

GETTIME,[v]

Returns the time lapse in second since Windows is started.

User can read the time into a user variable for later calculation.

GPIB,a,ss (Optional)

Sends string onto GPIB.

Where **a** - device address 1 to 31

ss - string to send.

Example: GPIB, 22,RESET

HALT

Stops program mode by generating an error. This command can be used to abort a script program in the event of error condition.

INBIT,p[,b,v]

Reads an I/O bit. (or byte) This command checks a specified bit (or byte) of the I/O (including the uncommitted I/Os provided for the user). The result will be displayed on the screen and can also be stored into a user variable.

Where p - Port 0 to 6
 b - Bit 0 to 7
 v - a pre-declared user variable.

Example: INBIT,0,1,STATUS
 will read port 0 bit 1 and store the result to variable STATUS

This command can also read the I/O port a byte at a time. If only 2 parameters are given, i.e. INBIT,p,v , the whole byte will be read and stored in the variable. The TTL I/O settings are stored in the TTLIO.INI file. You can edit this file to change port addresses.

See also: **OUTBIT**

INK[,p] (optional)

Inks the die with the optional inker hardware.

Where p - specifies what bin on the wafer map to ink.

Example: INK ;ink the current die
 INK,23 ;ink all die with bin#2 and 3

KEYFOCUS,p

Sets the keyboard input focus to either the command window or the video window. Since the keyboard arrow key functions differently depending on the input focus, this command can be used in a script file to control the input focus.

Where: p - C Command window
 V Video Window

Example: KEYFOCUS,V ; sets input focus to video window

KEystep,a,s

Sets jog key step.

The arrow buttons on the Jog windows are used for incremental motion. Its step sizes can be set using this command. They can also be set by typing them directly into the Jog Window.

Where: a - X,Y,Z,XY,ALL
s - step in user units [1..100]

Example: KEYSTEP,XY,10

Will set both the X and Y jog step to 10 user units.

LN,x

Moves the motorized turret to change the objective lens of the microscope.

Where: x - 1,2,3,4
or N next lens, P previous lens

Example: LN,2

MARK,p (Optional)

Marks characters by laser

Example: **MARK,"ABCD"**

Writes the character string ABCD on the panel.

MARK,Size=50

Sets the character dot size to 50 um (must NOT have any space character between the word Size and the '=' sign)

MARK,Speed=20

Sets the marking stroke speed to 20 um / dot. For example, given a dot size of 50um, the stroke speed would be $50 \times 20 = 1000$ um/sec.

Note:

When selecting the marking speed, you need to consider the laser's pulse repetition frequency to make sure the laser spot connects with the previous one to allow a continuous stroke.

MARK,Spacing=6

Sets character spacing to 6 dot sizes.

MARK,?

Queries marking parameters values.

Example response: **TM Size=50**
 TM Speed=50

 TM Spacing=6

MARK,Save=filename

Saves the current character set into file.

MARK,Load=filename

Loads a character set from a font file.

Note: Character sets are not designed to merge. If a new character set is loaded, the previous set will be discarded.

MCTR,xxx

Sends commands directly to the motor controller. This is a utility command used primarily for software debugging.

Example: MCTR,AX AN

Sends command AX AN directly to the OMS motion controller. (indexer)

ML,p

Moves to a preset location (test site). The preset locations can be set interactively using the Test Site page in the control panel or by loading a test site program.

Where: p - preset set location (test site) index. (1 to 999)

Example: **ML,0** ;Move to location 0

MOTIONDONE

Delays the execution of the program until current motion is completed.

This command is seldom used. Since this command is appended to all the motion commands, such as MV, MR etc., it is not necessary to use it explicitly, except in a few special situations.

MOVE,x,y (same as MV)

Moves absolutely to location x,y.

This command moves the current axes to the x and y position entered. The home position is always (0,0) and (x,y) is relative to this home position.

Example: MOVE,100,100 (move to location 100,100)

See Also: **MR**

MR,dx,dy

Moves relatively with the displacements of dx and dy.

This command moves the stage with reference to the current position.

Example: MR,10,10

See Also: **MV,MP**

MP,x,y

Moves to the physical position (x, y).

The physical coordinates refer to the stage reference that is the position of the negative limit switches. Please see XY Stage Basic 5.0 for an explanation of the physical coordinate system.

Example: MP,100,100

See Also: **MR,MV**

OUTBIT,p,b,v

Asserts or unasserts an I/O bit (including the uncommitted I/Os).

This is a utility command to be used only if you are familiar with the I/O port assignment.

Where p - port 0 to 6
 b - bit 0 to 7
 v - value 0 or 1

This command can also address an I/O port one byte at a time. If only two parameters are entered, e.g. OUTBIT,0,255, the entire byte will be addressed. In the above example, port 0 will be set to 255 by the command. The port address can be changed by editing the TTLIO.INI file.

See Also: **INBIT**

OVERDRIVE,d

Sets the edge sensor overdrive distance.

Where `d` - overdrive distance in Z axis unit

Example: OVERDRIVE,0.01

PATREC,p

Performs pattern recognition

Where `p` - parameter set (0 to 9)

See Also: **Pattern Recognition** section 7.1

PROBE,m

Enables or Disables probes.

Where `m` - mask for enabling probe.

Each probe is controlled by one bit of the mask, probe 1 by bit 0 (lsb), probe 2 by bit 1, probe 3 by bit 2, probe 4 by bit 3. A mask value of 1 will enable only probe 1. A mask value of 4 will enable only probe 3. To enable probes 1 and 3, the mask value should be 5. To enable all 4 probes the mask should be 15.

QUIT

Ends the ProbeMagic™ program.

READ232,p

Reads from an RS232 port.

Where `p` =
1 for COM1
2 for COM2
3 for COM3
4 for COM4

The ports read must be enabled by the configuration process before this command can be used.

Example: READ232,1
will read from COM1

See Also: **WRITE232**

ROTATE,a[,R]

Rotates the user coordinate system clockwise by theta degrees.

Where `a` - new theta in degrees

`R` –optional parameter is present indicates relative theta angle.

Example: ROTATE,0.5 ;software theta to 0.5 degree

ROTATE,0.5,R ;now software theta is 1.0 degree

See Also: **SETBASE**

RUN,fn

Retrieves and executes a ProbeMagic™ program written by the user. This command can execute a script program, a test site program or a map program.

Where fn - file name for the ProbeMagic to load and execute
WAFER to execute the currently loaded wafer map program
TEST to execute the currently loaded test site program

Example: RUN,DEMO1.SCR
will load and execute the DEMO1.SCR script program.
RUN,WAFER
will start running the currently loaded wafer program.

Note: Users can also load and execute a program using the main menu **File - Run Program**

SETBASE,[p]

Sets the baseline for user coordinate system. This command prompts the user to select two points. These two points define the baseline of the user coordinate system.

This is a very useful command as it takes the hassle out of using a mechanical theta alignment.

Where: p = M using motorized theta
Otherwise software theta rotation is used.

Please see discussion in **Section 6.9: Two Point Theta Alignment**

See Also: **ROTATE**

SCALE,a,s

Sets the scaling factor for the current axes.
Motion steps can be scaled to any convenient unit (mil,mm,cm,inch etc.). The scaling factor is the ratio of user units to microns.

Where a - X,Y,Z,XY,ALL
s - scale factor i.e. desired unit/micron

Example: SCALE, XY, 25.4 (user unit equals 25.4 microns i.e mil)

Note:

1. You can also change the scale by accessing the main menu (**Settings-Scale Factor**). It is much easier to change the scale using the menu as opposed to using the SCALE command.
2. Scale factors are stored in the configuration file PMAGIC.INI . The file is updated every time upon program exit, and reloaded when the program starts.

SEP,p[,l]

Sets the probe separate position.

Where p - separate position in user unit.
 l - lifts flag,
 if =1, the system will automatically lift the probes when XY tries to move
 with Z below separate position
 if =0, disables this feature

This is a probe protection mechanism. It prevents the probe from being damaged by accidental XY motion while the probe is making contact with the substrate. If the current probe position (Z position) is below the separate position, the XY axes will not move. If commanded to move, the error message, "Move Probe Up !", will be displayed on the status bar.

To disable this mechanism, set the separate position to a negative number. This mechanism is disabled by default.

Example: SEP,0.1,1

SETXY,x,y

Sets the current position to entered values. No motion occurs, with only the coordinate system being adjusted. This command affects the Preset Locations (test site) because it changes the home (0,0) position.

Where: x - becomes current x position
 y - becomes current y position

Example: SETXY,10,10; sets the current position to x=10 y=10

See Also: **ZERO**

SPEED,a,s

Sets the stage speed.

Where: a - X,Y,Z,XY,ALL

s - speed in user unit /sec

Example: SPEED,XY,5

When a=J, the command is used to set the Jog speed index. (1,2,or 3) should be entered for the Jog page in the Control Panel.

Example: SPEED,J,3

Note:

The speed can also be changed using the menu bar. Select **Edit, Parameters, Motion tab.**

SOT,[p]

Signifies Start of Test. When ProbeMagic™ is used in remote mode, this command informs the host.

Where: P - the optional parameter, if present can be **#d** or **#t**
#d - current die coordinates e.g. c(12,3)
#t - current test site name e.g. LOC#1

Example: SOT,#d ;signals start of test

See Also:**EOT**

STAGEREF

Moves to the bottom left corner of the stage and sets stage reference.

TOHOST,s

Sends message to remote host computer.

Where: s - string message

Example:TOHOST,LAST_DIE ;Sends "LAST_DIE" to host

WRITE232,p,ss

Writes string to a RS232 serial port

Where: p = 1 for COM1
2 for COM2
3 for COM3
4 for COM4
ss = any strings

Example: WRITE232,1,"Hello"
Writes the string "Hello" to COM1.

The COM port must be enabled in the configuration process before this command can be used.

See Also: **READ232**

ZERO[,a]

Zeroes the current axes.

Sets position counters to 0 .

Where a - axis X,Y,XY (default) ,Z,F,T

This command is a special case of the SETXY command. It is equivalent to SETXY,0,0

See Also: **SETXY**

ZMR,x

Moves the Z axis up or down. The default scale is in mm. The scale can be set using the SCALE,Z command or using the main menu.

Example: **ZMR,0.1** ;moves Z axis up 0.1 user unit.

ZMV,x

Moves the Z axis up or down to a position. The default scale is in mm. The scale can be set using the SCALE,Z command or using the main menu.

Example: **ZMV,0** ;moves Z axis to position zero

16.2 Batch commands

Batch mode commands can only be used in script files. They **cannot** be executed interactively or sent over via an interface. Here is a complete list in alphabetical order:

BEGIN

This command is placed at the start of a ProbeMagic™ script file. It comes after the subroutine and variable declarations. The instruction after BEGIN is the first one that is executed.

See Also: **END**

CALL,x

Calls a subroutine.

Where x - subroutine name

See Also: **SUB, RET**

CLOSE,p

Closes a user data file. .

Where p - IN or OUT

Example: CLOSE,IN ;closes the input file

See Also: **OPEN,WRITE**

DELAY[t,p]

Waits for user response or waits t milliseconds. This is a batch file command. If no parameter is given, the following message will be displayed in a window:

“Press Ok to continue, Cancel to Abort”

If "Ok" is pressed, the script file will continue running. If the “Abort” button is pressed, the script program will be terminated and the system goes back to the interactive mode. If the parameter “t” is given, the script program will suspend operation for t milliseconds. The parameter “p” is for you to specify a custom prompt message. If the “p” parameter is entered, the “t” parameter will be ignored. However, you must enter something for the “t” parameter, because the custom prompt “p” can only be the second parameter.

Example: DELAY,500 (waits half a second)
DELAY,-1,Press Ok after TEST101 is finished

If you want to change the position of the DELAY dialog, you can do so by moving it and then pressing Ok and Cancel. The dialog’s position information is saved to disk when the dialog closes. The next time the DELAY dialog is displayed, it retrieves the information from disk and restores it accordingly.

DISPN,xxxxx...

Displays a message in the Display Note window.

All string messages must be in double quotations “ “

Variables can also be displayed.

Example:LET,A=100
LET,COUNT=1

DISPN,"A= ",A, " Count=",count

will display:

A= 100.00 Count=1 in the Display Note window.

DISPN,CLEAR

Will clear the display note window

If you want to change the position and/or size of the Display Note window, you can do so by closing the window after adjusting its size or position. The window's size and position information is saved to disk when the window is closed. The next time the window is displayed, it retrieves the information from disk and restores accordingly.

See Also: **WRITE**

ECHO,p

Turns the command echo on the Command Window on or off.

Where: p - ON turn echo on (default)
 - OFF turn off command echoing
 - CLEAR clear the command window

END

This is the last line to appear in a ProbeMagic™ script file. This command pairs with the BEGIN command.

See Also: **BEGIN**

GETXY,vx,vy

Reads the current coordinates into two user variables vx and vy.

GOTO,p

Goes to a label in a ProbeMagic™ script file.

See Also: **Section 7.3 Script Program, Label**

IF THEN ELSE

IF command allows a batch file to conditionally execute a command.

For example,

IF condition THEN command ELSE command

or
IF condition THEN command

If ELSE is not present, then the next command will be executed if the condition fails.

Example:

```
IF A>15 THEN GOTO :GOOD ELSE GOTO :BAD
IF R>15 THEN GOTO :TOP
HOME
```

Note: Command nesting is not allowed. No IF THEN is allowed inside another IF THEN command.

INKEY,p,[s]

Reads a single digit number from the keyboard.

Where: p - a user defined variable used to hold the key stroke.

s - a string parameter holds a prompt to the user. If "s" is not given, the default prompt, "Please hit a key," will be used

Example: INKEY,a, Please enter acceleration

Will display "Please enter acceleration" and read the key into variable "a"

If you want to change the position of the INKEY dialog, you can do so by moving it and then pressing a key. The dialog position information is saved to disk when the dialog is closed. The next time that the INKEY dialog is displayed, it will retrieve the information from disk and restore accordingly.

See Also: **INPUT**

INPUT,p,[s]

Reads a number from the keyboard.

This number can have multiple digits and is terminated by pressing ENTER

Example: INPUT,%B

The optional "s" parameter is for a special prompt. If "s" is not given, the default prompt, "Please enter a number:", will be used.

Example: INPUT,%B,Please Enter the 2nd Value

Will display "Please Enter the 2nd Value" then store the input to %B

If you want to change the position of the INPUT dialog, you can do so by moving it and then entering the value. The dialog position information is saved to disk when the dialog is closed. The next time when the INPUT dialog is displayed, it retrieves the information from disk and restores it accordingly.

See Also: **INKEY**

LEND

Loop end. This command pairs with the LOOP command.

See Also: **LOOP**

LET

Assigns a value to a variable, where the value can either be a constant or a result from a math expression.

LET can also be used to assign a string to a string variable (\$) or convert a numerical variable to its string counter part.

Example:

```
LET, A=24
LET, B=A+6*E
LET, C=C+1
```

```
LET V$="Hello"
LET V1$=V$ + " Girl" ;V1$ will be "Hello Girl"
LET V2$=V$ + 123    ;V2$ will be "Hello123"
LET V3$=A           ;V3$ will be "24"
```

LOOP,n

Performs iterations in script programs.

LOOP & LEND commands are used for loop structures in a ProbeMagic script file. It is used only in batch mode.

Example:

```
..
LOOP,x
..
(ProbeMagic commands)
..
LEND
..
```


Where n determines the number of times that the loop executes; n can be a variable.

See Also: **LEND**

OPEN,filename,p

Opens a user data file for reading or writing. It requires the filename as a parameter. Only one data file may be opened at a time for input and output. MS-DOS has reserved the filename PRN for printer output. You may send a file to the printer by using PRN as a filename.

Where: p - IN for input
 - OUT for output
 - APP for append output

Example:

```
OPEN,TEST.DAT,IN
OPEN,RESULTS.OUT,OUT
```

READ,v1,v2..

Reads data from a previously opened ASCII file to one or several user variables.

Example:

```
READ,A
READ,B,C,D
```

There is no limit to how many variables can be read with a single READ command. However, the maximum command line length is fifty characters. There is no limit to the size of the user data file.

See Also: **WRITE**

READ\$,v\$

Reads string data from a previously opened ASCII file to a string variable. The string data is separated by commas ','.

Example:

```
VAR$,A$

BEGIN
OPEN,DATA.TXT,IN
```

READ\$,A\$

..

See Also: **READ,READLN\$**

READLN\$,v\$

Reads string data from a previously opened ASCII file to a string variable. The READLN\$ command will read until a new line character is encountered.

Example:

VAR\$,A\$

```
BEGIN
OPEN,DATA.TXT,IN
READLN$,A$
```

..

See Also: **READ\$,READ**

RET

Returns from a user subroutine.

See Also: **CALL, SUB**

SINGLESTEP,p

Turns the single step mode on or off.

Single step mode allows the user to step through a script file (or a section of a script program)

Where: p - ON or OFF

Single-step mode is used to debug a script file.

SUB,name

Declares a subroutine called name.

See Also: **CALL, RET**

VAR,x,..

Declares variable(s) for use by ProbeMagic™ script files. Must be placed before the BEGIN statement.

VAR\$,x,..

Declares string variable(s) for use by ProbeMagic™ script files. Must be placed before the BEGIN statement.

WRITE,ssss

Writes a message to a file that was opened by the OPEN command.

See Also: **DISP**

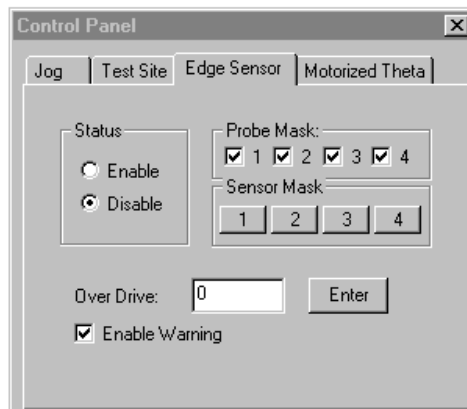
17.0 Add On Options

17.1 Probe Card Edge Sensor

The contact sensor protects from probe damage that could be caused by imperfect stage planarization. ProbeMagic™ supports the edge sensor that is usually built into the probe card. The sensing pins need to be connected as normally closed switches with one side pull to logically ground other pins connected to the sensing board in the computer. When contact is reached, a logic high should be sensed by the sensing board. ProbeMagic™ uses the computer's LPT2 port to sense the edge sensor switches. Pins 1,14,16 and 17 are the 4 sensing pins, Pin 25 is the ground pin, and the circuit current is approximately 1mA. Pins 6,7,8 and 9 provide 4 sets of enabling signals to the probe's motor driver.

When a probe card edge sensor is armed and triggered, the software will stop all downward Z motion. Users can program in an **Overdrive** value to ensure good probe contact with the device undergoing testing. Beware!! The contact sensing may be **Disabled** after ProbeMagic™ is started up. If you have used the program, make sure to use the CSENSOR, ON command before the Z down commands. If you drive ProbeMagic™ with a host computer, be sure to send over the CSENSOR, ON command before probing.

When the option is installed, the Control Panel will have an extra page titled **Edge Sensor**.

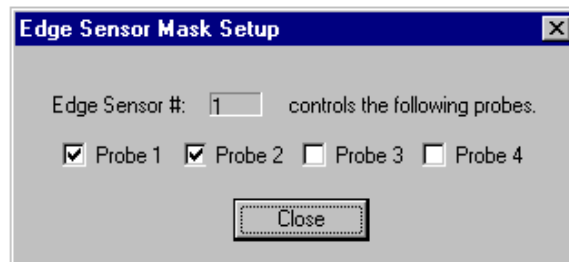


With this option, you can interface up to 4 edge sensors. They can be grouped into 4 groups (1 or 2 probes in one group) with each edge sensor controlling one group of probes.


In large area probing, the planarity is hard to maintain. The use of multiple edge sensors is particularly useful when the application requires probes to be placed widely apart from each other. This option allows each probe to have its own edge sensor. The touch down motion of the probe will stop when the corresponding edge sensor is triggered.

This option also allows for probes to be independently enabled and disabled, giving way to independent probe motion. This means that you do not always have to move all the probes up and down together. In the **Edge Sensor** page, the **Probe Mask** check boxes enable and disable each group of probes. When a Z move (ZMV, or ZMR) command is issued, only the probes that are enabled move. You can also enable or disable probes using the PROBE command. You can use the PROBE command in a script file to programmatically enable or disable probes.

Edge sensors can also be individually configured to control 1 to 4 probes. This is necessary if not all the probes have edge sensors. For example, if you have 4 probes and have edge sensors attached to probes 1 and 3, and probe 2 is placed very close to probe 1, as well as probe 4 very close to probe 3, then you should use the edge sensor on probe 1 to control both probes 1 and 2. Similarly, the edge sensor on probe 3 should be used to control both probes 3 and 4. In this way, during a touch down motion, edge sensor 1 will stop both probes 1 and 2 and edge sensor 3 will stop both probes 3 and 4. Although you can configure a probe to be controlled by more than one edge sensor, this practice is not recommended. When a probe is controlled by more than 1 edge sensor, triggering any controlling edge sensor will stop the probe. By default, edge sensor 1 controls probe 1, sensor 2 controls probe 2, etc. You can change the probe to sensor assignment by pressing the buttons in **Sensor Mask**. The **Sensor Mask** buttons launch dialogs for assigning probes to edge sensors:



The above dialog shows that edge sensor 1 will control probes 1 and 2.

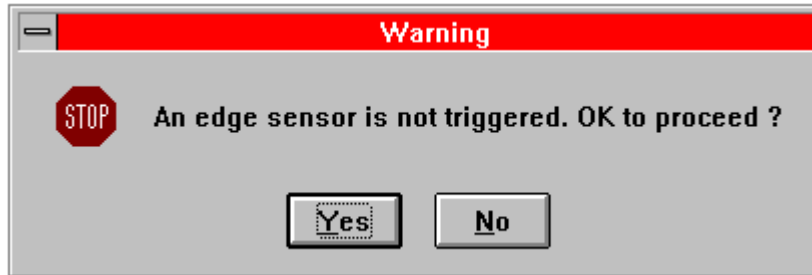
The **Status Enable/Disable** radio buttons on the **Edge Sensor** page can globally enable or disable all edge sensors together. (Toggling the edge sensor button  on the **Jog** page has the same effect.) The CSENSOR command can be used to set the global Enable/Disable status. Users can implement the CSENSOR command in a script file to programmatically enable or disable edge sensors.

The status of the edge sensors and probes are shown at all times at the right end of the status bar using color bars. The probe status is shown on the top half of the bar: red meaning enabled and gray meaning disabled. The edge sensor status is shown on the bottom half on the bar: red meaning enabled and triggered, black meaning enabled but not triggered, and finally, gray meaning disabled.

The overdrive value is common to all probes with edge sensors. During a downward Z motion, the software continually checks all the edge sensors. If an edge sensor senses a contact, the computer will immediately stop the motion of the assigned probe (or group of probes). After

all the enabled edge sensors are triggered, the software will then start the overdrive motion. The touch down motion is completed in two stages; 1:move to contact, 2:overdrive.

The software expects that all the enabled edge sensors will be triggered during a touch down action. It will generate a warning message if any of the edge sensors are not triggered.



The warning will pause any program (script, map, or test site) and give you a chance to make sure that the edge sensors are properly connected. This feature is enabled upon software start up. This feature may be disabled using the command CSENSOR, NOWARN. The CSENSOR, WARN command will re-enable this warning feature.

The software will prohibit any horizontal motion if any of the enabled edge sensors are triggered. The computer will beep and the message, "Move probe up !," will be displayed in the status bar when any horizontal motion is attempted while an edge sensor is triggered. Optionally, you can enable an auto probe lift feature. The feature is enabled using the SEP command. For example, SEP,0.1,1 will set a **separate distance** of 0.1 mm and enable auto probe lift. With this feature enabled, any attempt to move XY while an edge sensor is triggered will cause the probes to lift the separate distance (in this case is 0.1 mm).

The position of each probe is NOT tracked individually. There is only one electronic counter that tracks the overall Z position. From time to time, you will need to re-synchronize the probe positions. The following is a list of commands that will do that:

```
CSENSOR, ON           ;turns contact sensing on
OVERDRIVE, 0          ;sets overdrive to zero
ZMR,-0.5              ;moves Z down, all probes will stop at contact
ZERO, Z               ;zeros the Z position counter
..
(reset normal overdrive value, and start normal probing)
..
```

The following commands are related to the Edge Sensor option:

CSENSOR,s - Enables or Disables contact sensing.
Where: s can be ON, OFF, WARN or NOWARN
Example: CSENSOR,ON

OVERDRIVE,d- Sets edge sensor overdrive distance.

Where: **d** is overdrive distance in Z axis unit
Example: OVERDRIVE, 0.01

PROBE,m

- Enables or Disables probes.

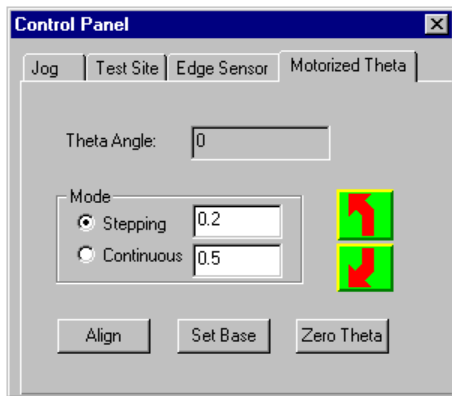
Where: **m** is the mask for enabling probe.

Each probe is controlled by one bit of the mask, probe 1 by bit 0 (lsb), probe 2 by bit 1, probe 3 by bit 2, probe 4 by bit 3. A mask value of 1 will enable only probe 1. A mask value of 4 will enable only probe 3. To enable probes 1 and 3, the mask value should be 5. To enable all 4 probes, the mask should be 15.

17.2 Motorized theta

The motorized theta is another add-on option for the LCD2424. It motorizes the micrometer that controls the stage theta. When this option is installed, the function of the SETBASE command will change. The command will move the micrometer to adjust for theta. (If no motorized theta is installed, the SETBASE command will rotate the coordinate system. Please refer to section 6.71 Software theta.)

When the option is installed, the Control Panel will have an extra page named **Motorized Theta**.



The following command pertains to the motorized theta:

ALIGN,[m]

- Aligns substrate.

Where **m** = **M** is the motorized theta that is to be used to adjust theta; otherwise, the software theta rotation will be used.

See Also: **SETBASE**

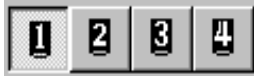
Note: **ALIGN** - Aligns substrate by referencing the alignment marks
SETBASE - Aligns substrate by a straight line of the substrate.

TMR,d - Theta moves relative
Where d = the distance moved

TMV,p -Theta moves absolute
Where p = the position to move to

17.3 Motorized Turret

The motorized turret option (MT405) motorizes the microscope objective lens turret. It is an automatic objective change. It is an especially useful option if your application requires frequent magnification change. With the MT405, you can change the objective either by selecting from



the tool bar button or from menu bar, **View, xxX** (e.g. 10X) or by using the LN command.

The LN command has the following syntax:

LN,x - Changes Objective Lens

Where: x = 1,2,3,4 each correspond to one objective

N - for next objective

P - for previous objective

ProbeMagic™ communicates with the MT405 via COM2 port of the computer. The controller for the MT405 may be integrated into the AF405 controller when both options are ordered.

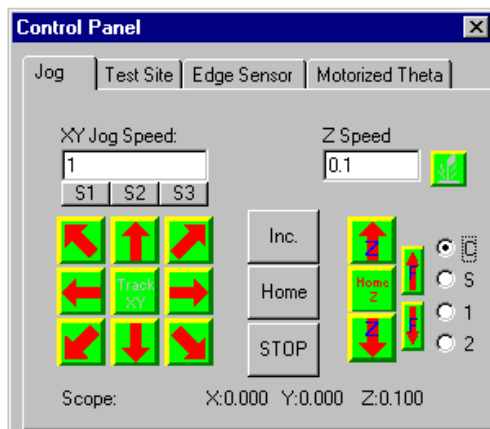
17.4 Motorized Microscope & Positioners

For customers who wish to motorize the probe station's microscope XY transport and/or the probes, can purchase our MC300 motor controller and motors. A special extension DLL named MC300.DLL is written to control the motorized micropositioner and microscope. The extension DLL can control up to four micropositioners and one microscope transport from the ProbeMagic™ software.

Both the micropositioners and the motorized scope transport are driven by the MC-300 controller. The MC-300 is a three axes stepper motor controller with daisy chain capability that allows up to ten MC-300s to be controlled via one RS232 COM port. In the following text, one micropositioner or motorized microscope transport will be generically referred to as a channel. Each channel has three axes, X, Y and Z. Each MC-300 controls one channel.

User Interface

When the MC300.DLL is loaded, additional controls will appear on the Jog page in the control panel.



The radio buttons **S,C,1**, and **2** select which channel the Jog arrows controls. Each letter or number represents a different component. **C** refers to the wafer chuck XY table, **S**, the microscope XY transport, **1**, probe #1, and **2**, probe #2. At the bottom of the Jog page, the position of the MC300's current channel will be displayed. It may refer to the position of the microscope or any one of the probes.

Command Added

The MC300.DLL adds the following commands to ProbeMagic™. The commands can be used in ProbeMagic™ script programs or typed into the Interactive Input Window for immediate execution.

The first parameter of all the commands, except PLL, specifies the channel. The following characters may be used:

- S** - Scope
- A** - All Channel
- 1** - Positioner 1
- 2** - Positioner 2.
- 3** - Positioner 3.
- 4** - Positioner 4.

PLL,s

Low Level Command

Where **s** - sends directly to the MC-300 controller. The

PROBEDLL.DLL do not process the **s** command.

Example: PLL,0ZE

PMV,c,x,y Moves Absolutely
Where: c - channel
x,y - destination coordinates.

Example: PMV,1,1.25,4.50

PMR,c,dx,dy Move Relatively

Example: PMR,S,1,1

PZM,c,z Moves Z axis absolute

Example: PZM,A,0

PZR,c,dz Moves Z axis relative

Example: PZR,A,100

PLT Lifts all probes below separate position to their separate positions

PMP,c,p Moves to preset

Example: PMP,A,1

PSP,c,p Sets current XY position to preset p

Example: PSP,1,3

PALIGN,c Aligns channel
Determines the software theta rotation

Example: PALIGN,1 ;Aligns Probe 1

PCALI,c Calibrates channel
Determines the error compensation factors

Example: PCALI,2 ;Calibrates probe 2

Coordinate Systems

The micropositioners' coordinates have the same scale as that of the XY stage. The MC300.DLL reads the scale factors from ProbeMagic™ during initialization so that PMV,1,10,10 moves probe 1 to position (10,10) on the user panel. If a different scale factor is

selected (e.g. to mm, mil or um) for ProbeMagic™ positioners, scale factors for the XY stage will also be changed accordingly. However, since MC300.DLL only reads the scale factor during initialization, ProbeMagic™ will need to be restarted for the scale change to take effect in the MC300.DLL.

It is recommended that all of the positioners be set to the home position (0,0) so that they appear at the center of the screen with the probe tips lining up with the video cross hair. This will give all of the positioners the same home reference. It is also recommended to orient the XY axes of the positioner roughly parallel to that of the XY stage.

Since the mechanical axis orientation of each positioner may be different from that of the XY stage, PROBEDLL.DLL uses software theta to translate the coordinates before sending commands to the MC-300. The translation process is completely transparent to the user. With the software theta correctly setup, users only need to consider one coordinate system. Also, there is no need to mechanically orient the axes of the positioner exactly to that of the XY stage.

The command PALIGN initiates the process to determine the theta offset between the selected positioner and the XY stage.

To have one common coordinate system for all channels, including the XY stage, is essential for high-level features such as **Probe Tracking** and **Drag and Drop Probe Navigation**.

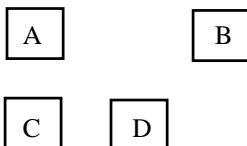
To achieve high accuracy, the MC300.DLL uses software error compensation. The command, PCALI,c, initiates a process to determine the error compensation factors. This process calibrates the accuracy of the positioner with respect to the XY stage accuracy. The process is similar to that which is initiated by the CALISTAGE command.

Programming Tips on Using Presets

The MC-300 uses a daisy chain technique to allow one COM port to control up to 10 micropositioners. Users can send commands to all micropositioners simultaneously or to each individually. To maximize output, users should utilize the broadcast commands that move all of the micropositioners on the daisy chain together. To broadcast a command to all positioners, use the channel 'A' (All) for the channel parameter in the command.

For example:

The command PMV,A,10,10 will move all of the positioners on the chain to location (10,10). The broadcast command is especially meaningful when used with the PMP (move to preset) command. PMP,A,1 will move all of the micropositioners to their preset location 1. Since each positioner has its own set of preset locations, the positioners will move to different places.



E

For example, if one circuit has the above pad positions, you will need to conduct three measurements, one across pads A and B, one across pads C and D and then across pads C and E. Two micropositioners are used to carry out the probing. You may utilize the preset feature as follows: For the positioner on the left, set preset location 1 where the probe tip is, on pad A, set its preset location 2 to be where the probe tip is on C. Also set its preset location 3 at the same place. For the positioner on the right, set its preset location 1 on pad B, preset location 2 on pad D and preset location 3 on pad E. In this way, the following script command sequence will move the probes to the appropriate places for the measurement.

```
...
PMP,A,1      ;probe pads A and B
..           ;makes measurement 1
PMP,A,2      ;probe pads C and D
..           ;makes measurement 2
PMP,A,3      ;probe pads C and E
..           ;makes measurement 3
..
```

Remember the broadcast commands will be received and executed by all the positioners on the daisy chain. Thus, you must beware of its side effect. In the above case, if the system has more than two positioners, you need to set preset locations 1,2 and 3 of the other positioners all at the same place so that they will not move when they receive the PMP,A,1 or PMP,A,2 and PMP,A,3 commands.

Although the control dialog only displays 10 preset locations on its list box, the MC-300 supports up to 50 preset locations that users can use through the PSP command to set preset location numbers 10 to 49.

Configuration Remarks

The section explains the possible impacts on the ProbeMagic™ configuration when using the MC300.DLL with MC-300.

1. Please distinguish the Z axis control by ProbeMagic™ directly (which sometimes is referred to as the Platen Z axis) with the Z axes on each micropositioners (which is controlled by the MC300.DLL). Both the Platen Z axis and the micropositioners's Z axes move the probes up and down, from and to the pads. There is only one platen Z axis which is controlled by the Z arrows in the **Jog** page of the control panel and with the ZMV and ZMR script commands. The Platen Z axes can drive multiple motors. On the other hand, the Z axes on the micropositioners can be controlled using the PZM and PZR commands. Each micropositioner's Z axis drives only one motor. A probe station must have either or both sets of the Z axes.

2. For systems without the Platen Z axis, it should be configured with the flag, **NoPlatenZ**, set to TRUE. With this configuration, the Z axis arrows in the **Jog page** will be grayed out and disabled. The commands, ZMV and ZMR, will have no effect. The Z position displayed on the status bar will display the word 'Absent'.
3. Edge sensors can be attached to both the Platen Z axis or the Z axes on the micropositioners. However, they are controlled and configured from different places in the software. The edge sensor for the Platen Z axis is controlled by the edge sensor dialog, which can be launched by selecting **Edge Sensor** from the **Option** menu. This dialog has NOTHING to do with the edge sensor on each MC-300 controlled micropositioner. The edge sensors on the micropositioners are controlled by the PLL,A,OD command which also sets the overdrive value for the edge sensor.

MC300.DLL Configuration

The software uses MC300.INI to save configuration information. During setup, users may need to edit the file to change the information. The MC300.INI can be edited using any text editor.

The parameters that you are most likely to change are **COMSettings**, **JogDelayHigh** and **JogDelayLow** in the [Main] section. The COMSettings specifies the COM port configuration. The two JogDelay parameters determine the High and Low Jog speed of the jog arrows. The larger the delay factor value, the slower the jog speed. The minimum value is about 20.

```
[Main]
COMSettings=COM1:9600,n,8,1
JogDelayHigh=30
JogDelayLow=100
```

```
[Chain0]
ChannelID=0
IsScope=1
Theta=0
Shew=0
AlignR=1000
CaliX=1000
CaliY=1000
XErrComp=1
XumScale=1
XScale=1000
XPos=0
XDir=1
YErrComp=1
YumScale=1
YScale=1000
YPos=0
YDir=1
ZErrComp=1
ZumScale=1
ZScale=25.4
ZPos=0
ZDir=1
```

..
...
..Etc

17.5 KLA Data File Interface

Introduction

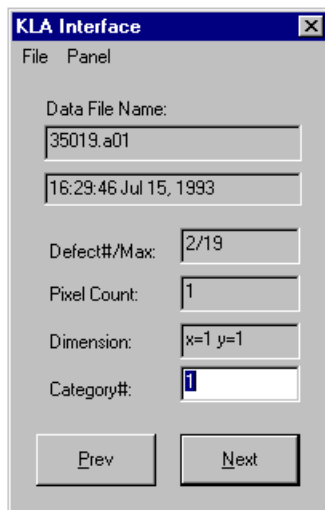
The KLA Data File Interface is a ProbeMagic™ extension DLL designed to read and interpret the data generated by the KLA Acrotec 6000 series LCD panel inspection system. This interface greatly facilitates panel repair by utilizing the defect coordinate data generated by the inspection machine.


Software Installation

The interface is implemented in, **kla_intf.dll**, the DLL file located in the ..\ExtDLL sub-directory. (Usually C:\Pmagic\ExtDll). The following steps load the interface into the ProbeMagic™ software:

1. In the command window, type **Config** <Enter>
2. Enter the password to launch the **System Configuration** dialog.
3. Select the **Options** page of the dialog.
4. In the **External Options** block, **Available Options** list, select **kla_intf.dll**.
5. Press the **Add** -> button.

User Interface



After the option is installed, the button  will appear in the button bar. Pressing the button launches the following dialog:

There are several text boxes on the dialog; they display the defect information. There are two buttons; the **Next** button moves the microscope to the next defect location. Pressing the enter key after entering the new **Category#** number has the same effect as pressing the **Next** button. The **Prev.** button moves the stage back to the previous defect location.

These are the following menu selections:

File

Load Data File - Launches a file; opens dialog to load in a data file.

Write Data File - Launches a file saved as dialog to create an output data file containing the new classification number.

Panel

Align Panel - Moves the microscope to the left and right alignment marks to align the panel

Go to Defect - Moves to a specified defect location

Panel repair operation steps

1. Place the panel onto the stage.
2. Load the corresponding KLA data file by selecting from the menu, **File, Load Data File**.
3. Align the panel by selecting from the menu, **Panel, Align Panel**. The microscope will move near the left alignment mark. The following dialog will display:



Users should move the microscope using the Video Navigator or the Control Panel of the ProbeMagic™ so that the video cross hair lines up with the left alignment mark on the panel. Then press the **OK** button. The microscope will move close to the right alignment mark of the panel and the following dialog will be displayed:



Users should line up the cross hair and then press **Ok**.

Note: You may need to do this alignment process 2 to 3 times to get the panel completely aligned.

4. After the panel is aligned, users can go to each defect consecutively by pressing the **Next** button. At the defect site, the user can review or repair it and then key in a new classification number in the **Category#** edit box.

- After all of the defect sites have been reviewed/repared, the user may create an output data file by selecting from the menu **File, Write Data File**.

Sample Data File Listing

Note: The line number and the comments are not part of the KLA data file.

```

1: 1 ;Panel Number
2: 16:29:46 Jul 15, 1993 ;Data creation data and time
3: 19 ;Total number of defects
4: 0 ;Total number of unclassified defects
5: 2, 10, 0, 0, 0, 0, 7, 0, 0, 0, ;Line 5 to 14, number of defects for
6: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ;classification 0 - 99
7: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
8: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
9: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
10: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
11: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
12: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
13: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
14: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
15: 7.431e-01 ;Total defect density
16: 0.000 ;Density of unclassified defects
17: 7.431e-01 ;Line 17 to 26, density of classified defect for
18: 0.000 ;groups 0-9,10-19,20-29 ... up to 90-99
19: 0.000
20: 0.000
21: 0.000
22: 0.000
23: 0.000
24: 0.000
25: 0.000
26: 0.000
27: 57.365,73.305 ;Defect orgin relative to front left corner of the
;glass in mm
28: 10.820,73.280 ;left alignment marks relative to front left corner
29: 57.535,73.530 ;right alignment marks relative to front left corner
30: 0,1,-6.020,-6.665,398,75,27,6,???,0,0 ;Line 30 to the end of the file are
31: 1,1,-16.270,-4.235,1,1,1,1,???,0,0 ;individual defect data
32: 2,1,-23.195,-3.735,174,21,31,0,???,0,0 ;Where
33: 3,1,-24.435,-4.065,14,6,3,1,???,0,0 ;Column 1 = Defect Number
34: 4,1,-28.290,-6.155,4,2,2,1,???,0,0 ;Column 2 = Display Number
35: 5,1,-30.775,-6.285,2,1,2,1,???,0,0 ;Column 3 = X location relative to origin
36: 6,1,-4.065,-16.480,7,2,4,6,???,0,0 ;Column 4 = Y location relative to origin
37: 7,1,-37.850,-19.705,6,2,4,1,???,0,0 ;Column 5 = Defective Pixel Count
38: 8,1,-2.610,-21.470,1,1,1,6,???,0,0 ;Column 6 = X dimension of defect
39: 9,1,-21.430,-28.925,92,54,30,1,???,0,0 ;Column 7 = Y dimension of defect
40: 10,1,-26.835,-31.540,5,3,3,1,???,0,0 ;Column 8 = Classification number
41: 11,1,-33.600,-28.030,4,1,4,1,???,0,0
42: 12,1,-28.590,-36.895,27,13,4,1,???,0,0
43: 13,1,-12.240,-43.700,2,1,2,1,???,0,0
44: 14,1,-6.575,-49.710,89,11,14,6,???,0,0
45: 15,1,-11.025,-50.990,258,18,20,6,???,0,0
46: 16,1,-40.825,-46.715,417,23,38,0,???,0,0
47: 17,1,-3.895,-53.955,48,10,7,6,???,0,0
48: 18,1,-39.040,-56.140,1,1,1,6,???,0,0

```


17.6 OrboTech Data File Interface

Introduction

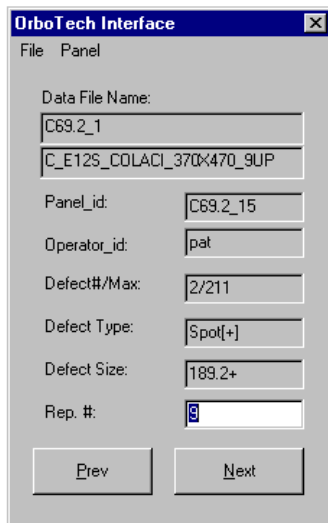
The Orbotech Data File Interface is a ProbeMagic™ extension DLL designed to read and interpret the data generated by the Orbotech LCD panel inspection system. This interface greatly facilitates panel repair by utilizing the defect coordinate data generated by the inspection machine.


Software Installation

The interface is implemented in the file, **orbotech.dll**, the DLL file located in the ..\ExtDLL sub-directory. (Usually C:\Pmagic\ExtDll). The following steps load the interface onto the ProbeMagic™ software:

1. In the command window, type **Config** <Enter>
2. Enter the password to launch the **System Configuration** dialog.
3. Select the **Options** page of the dialog.
4. In the **External Options** block, **Available Options** list select **orbotech.dll**.
5. Press the **Add ->** button.

User Interface



After the option is installed, the button  will appear in the button bar. Pressing the button launches the following dialog:

There are several text boxes on the dialog; they display the defect information. There are two buttons; the **Next** button moves the microscope to the next defect location. Pressing the enter key after entering the new **Rep. #** number has the same effect as pressing the **Next** button. The **Prev.** button moves the stage back to the previous defect location.

There are the following menu selections:

File

Load Data File - Launches a file; opens dialog to load in a data file.

Write Data File - Launches a file saved as dialog to create an output data file containing the new classification number.

Panel

Align Panel - Moves the microscope to left and right alignment marks to align the panel

Go to Defect - Moves to a specified defect location.

Panel repair operation steps

1. Place the panel onto the stage.
2. Load the corresponding Orbotech data file by selecting from the menu, **File, Load Data File**.
3. Align the panel by selecting from the menu, **Panel, Align Panel**. The microscope will move near the left alignment mark. The following dialog will be displayed:



Users should move the microscope using the Video Navigator or the Control Panel of ProbeMagic™ so that the video cross hair lines up with the left alignment mark on the panel, and then press the **OK** button. The microscope will move close to the right alignment mark of the panel and the following dialog will be displayed:



User should line up the cross hair, and then press **Ok**.

Note: You may need to do this alignment process 2 to 3 times to get the panel completely aligned.

4. After the panel is aligned, the user can go to each defect consecutively by pressing the **Next** button. At the defect site, the user can review or repair the defect site and then key in a new classification number in the **Rep.#** edit box.
5. After all the defects have been reviewed/repared, the user may create an output data file by selecting from menu **File, Write Data File**.

Samples Data File Listing

```

panel_type      = "C_E12S_COLACL_370X470_9UP"
panel_id       = "C69.2_15"
operator_id    = "pat"
queue         = "learn"
scan_status   = "TERM OK"
go_repeats    = ""
nogo_repeats  = ""
image_file_prefix = ""
slot_no       = 1
num_repeats   = 9
glass_size.x  = 370.000
glass_size.y  = 470.000
board_center.x = 188.000
board_center.y = 232.000
panel_status  = "NOT CLASSIFIED"
align_targets = (-131.317 212.885) (132.682 212.885) (-131.317 -221.115)
num_defects   = 211
defect_list   = "# Rep no. X coord. Y coord. Size Retype Type Cell Zone Size-X Size-Y Class ImageSuffix Row_Cell_Num Col_Cell_Num"
3, -91.573, 136.041, "189.6+", "Cut[+]", "Cut", "-", "169.3+", "181.2+", "-", "", 154, 181,
9, 148.566, 116.957, "189.2+", "Spot[+]", "Spot", "-", "181.8+", "175.0+", "-", "", 95, 236,
9, 148.529, 116.719, "166.7+", "Short[+]", "Short", "-", "181.8+", "162.5+", "-", "", 95, 235,
3, -137.028, 146.323, "159.0+", "Dust[+]", "Dust", "-", "119.1", "181.2+", "-", "", 186, 41,
2, -139.323, -23.481, "154.6+", "Short", "Short", "-", "181.8", "168.8+", "-", "", 88, 34,
4, -30.335, -173.425, "151.7+", "Spot", "Spot", "-", "181.8+", "175.0+", "-", "", 52, 27,
3, -91.473, 136.292, "142.5+", "Cut", "Cut", "-", "181.8+", "181.2+", "-", "", 155, 181,
8, 137.097, -21.197, "129.2+", "Short", "Short", "-", "181.8+", "131.2", "-", "", 95, 200,
4, -20.387, -162.507, "120.7", "Short", "Short", "-", "119.1", "168.8", "-", "", 86, 58,
6, 17.638, 170.946, "115.8+", "Spot", "Spot", "-", "175.5+", "156.2", "-", "", 262, 175,
9, 100.275, 156.100, "114.4+", "Spot", "Spot", "-", "150.4+", "162.5+", "-", "", 216, 87,
6, -32.483, 169.270, "109.9", "Cut", "Cut", "-", "119.1", "131.2", "-", "", 256, 21,
3, -96.618, 104.907, "102.9", "Short", "Short", "-", "94.0", "156.2", "-", "", 58, 165,
9, 148.993, 117.019, "101.3", "Protrusion", "Protrusion", "-", "106.6", "137.5", "-", "", 96, 237,
4, 28.401, -103.972, "100.9", "Short", "Short", "-", "144.2", "112.5", "-", "", 266, 208,
1, -98.684, -107.383, "99.4", "Protrusion", "Protrusion", "-", "119.1", "100.0", "-", "", 255, 159,
8, 90.581, 1.197, "98.8", "Short", "Short", "-", "131.6", "118.8", "-", "", 164, 57,
5, -10.913, -51.716, "91.8", "Short", "Short", "-", "94.0", "118.8", "-", "", 1, 87,
9, 148.993, 116.807, "91.4", "Protrusion", "Protrusion", "-", "125.4", "156.2", "-", "", 95, 237,
7, 81.114, -134.620, "89.8", "Protrusion", "Protrusion", "-", "106.6", "106.2", "-", "", 171, 28,
8, 114.721, -15.473, "83.5+", "Short", "Short", "-", "150.4+", "131.2", "-", "", 113, 131,
5, -21.224, -0.795, "80.9", "Protrusion", "Protrusion", "-", "106.6", "106.2", "-", "", 158, 55,
1, -85.597, -188.534, "75.9", "Protrusion", "Protrusion", "-", "94.0", "93.8", "-", "", 6, 199,
2, -134.140, 1.103, "70.5", "Protrusion", "Protrusion", "-", "69.0", "106.2", "-", "", 164, 50,
1, -107.473, -94.399, "69.1", "Short", "Short", "-", "94.0", "93.8", "-", "", 295, 132,
7, 121.233, -147.773, "68.6", "Short[+]", "Short", "-", "69.0", "56.2", "-", "", 131, 151,
3, -93.448, 99.576, "68.6", "Protrusion", "Protrusion", "-", "81.5", "81.2", "-", "", 42, 175,
5, 26.903, 21.137, "66.8", "Protrusion", "Protrusion", "-", "81.5", "81.2", "-", "", 226, 203,
1, -135.933, -168.687, "66.5", "Protrusion", "Protrusion", "-", "81.5", "87.5", "-", "", 67, 44,
8, 137.324, -21.308, "65.8+", "Protrusion[+]", "Protrusion", "-", "181.8+", "156.2", "-", "", 95, 201,
6, 10.201, 193.107, "65.0", "Island", "Island", "-", "81.5", "93.8", "-", "", 330, 152,
2, -101.684, 9.586, "64.6", "Short[+]", "Short", "-", "94.0", "68.8", "-", "", 190, 150,
9, 77.109, 175.272, "60.4", "Protrusion", "Protrusion", "-", "94.0", "56.2", "-", "", 275, 16,
6, -20.855, 98.355, "60.4", "Protrusion", "Protrusion", "-", "81.5", "75.0", "-", "", 38, 56,
2, -125.894, 50.781, "59.7", "Short[+]", "Short", "-", "69.0", "56.2", "-", "", 317, 75,
8, 125.411, 55.124, "58.7", "Island", "Island", "-", "56.4", "81.2", "-", "", 330, 164,
..
..
..

```


17.7 Spectra-Physics Laser Interface

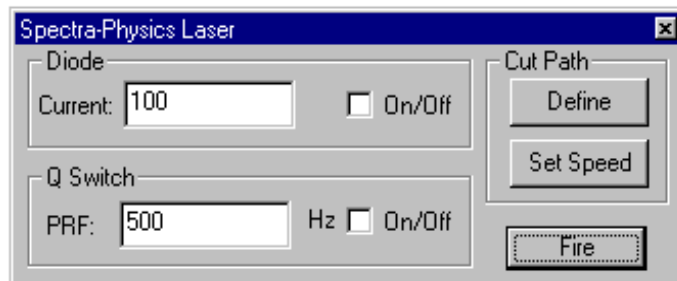
Introduction

ProbeMagic™ interfaces via RS232 to Spectra-Physics T-Series lasers. All the laser parameters including, diode current, Q switch PRF, laser energy level, etc., are controlled via the software. ProbeMagic™ provides a special dialog box for interactive control of laser parameters in ProbeMagic™. Users can also control the laser programmatically using the script commands.

The laser interface is implemented as an external option (splaser.dll). The splaser.dll is normally located in the ..\extdll directory e.g. c:\pmagic\extdll\. You can load the option into ProbeMagic™ by following these directions:

1. In the command window, type **Config** <Enter>
2. Enter the password to launch the **System Configuration** dialog.
3. Select the **Options** page of the dialog.
4. In the **External Options** block, **Available Options** list select **splaser.dll**.
5. Press the **Add ->** button.

When the option is loaded, the button  will appear on the button bar. Pressing the button will launch the laser control dialog. User should only need to load the option once. When closing, ProbeMagic™ remembers which external options were loaded and will load them automatically the next time ProbeMagic™ is started.



There are three sections on the dialog, the **Diode**, **Q Switch** and **Cut Path** sections:

In the **Diode** Section, the laser diode current can be set in hundredths of amps, as well as the control diode being turned On/Off with a check box.

The **Q Switch** section controls the Q-switch parameter, PRF (pulse repetition frequency) and the Q-switch On/Off.

The **Cut Path** Section has two buttons:

Define - Allows users to define a path for the laser to cut. The software prompts the user to select an ending point and then a starting point. It will fire the laser in Pulse or CW mode along the cut path.

Set Speed - Sets the speed of travel in path cutting. The laser can cut in either CW or Pulse mode. In pulse mode, the Q switch PRF and the speed of cutting defines the pulse overlap of a cut.

Finally, the **Fire** button pulses the laser one shot.

Controlling the laser from a script program

The splaser.dll, which implements the laser option, adds 3 commands to ProbeMagic™. The commands can be typed in and executed interactively from the **Command Windows** or used in script files to execute programmatically.

SP - Sends Spectra-Physics Laser commands.

This command allows the sending of any Spectra-Physics laser command from ProbeMagic™. A listing of all the Spectra-Physics laser commands can be found in the Spectra-Physics laser manual section 5-5. To send any Spectra-Physics laser commands from ProbeMagic™, the user needs to prefix the command with SP. For example, to set laser the PRF to 900 Hz using the Q command from ProbeMagic™, the command will be:

SP,Q:900<Enter>

Where <Enter> means pressing the Enter key on the keyboard.

To turn on the Q switch:

SP,G1<Enter>

The user can also execute status request commands. The reply from the laser will display in the **Command Windows**. Any reply message from the laser will have a prefix of **L:** when displayed in the command windows.

For example, to request laser system status flags, the command would be:

SP,?S<Enter>

The reply may be:

L:032

The following script program will cut a square pattern:

```
;CUT_SQ.SCR
;The script program demo cutting a
;square pattern using the
;spectra-physics Model 7000 series laser

BEGIN
SP,Q:1000 ;PRF 1KHz
SP,C1:5000 ;Current 500 mA
SP,D1 ;Laser diode on
SP,G1 ;Q-switch on
MR,0.2,0 ;Moves in a square pattern
MR,0,0.2
MR,-0.2,0
MR,0,-0.2
SP,G0 ;Laser off
END
```

LLOAD,fn - Loads laser settings file from disk, Where fn - filename

LSAVE,fn - Saves laser settings file to disk, Where fn - filename.

Get the most out of the laser option

Together with other ProbeMagic features, the laser option enhances the functionality of the laser tremendously. Here we use three application examples to illustrate:

1. One button parameter switching

In this laser application on the substrate, there are three different materials, Al, ITO and Cr. Each material requires a different set of laser parameters.

ProbeMagic™ allows users to define buttons on the button bar to execute a script program. To facilitate parameter switching, one can store three sets of laser parameters (corresponding to cutting Al, ITO and Cr) into three laser setup files named AL.LSU, ITO.LSU and CR.LSU, and then create three buttons:

- has script file with command LLOAD,AL.LSU
- has script file with command LLOAD,ITO.LSU
- has script file with command LLOAD,CR.LSU

In this way, one push of the button changes the complete set of laser parameters, such as laser power, PRF etc.

2. One button special cut pattern.

In this application, the user wants to be able to control 4 laser zaps to isolate a defective transistor from circuit with one push of a button. You can create a script file that executes the cut pattern.

For example, a script file would be:

```
SUB,FIRE
SP,G1           ;Fires Laser one pulse
DELAY,10
SP,G0
RET

BEGIN
SP,Q:10        ;Sets laser to 10Hz
CALL,FIRE
MR,0.1,-0.1
CALL,FIRE
MR,-0.1,-0.1
CALL,FIRE
MR,-0.1,0.1
CALL,FIRE
END
```

You can associate this script file with a button on the button bar. With one push of the button, it will cut the pattern. Given the flexibility of script files, you can cut any pattern with any number of laser zaps.

The splaser.ini file

The laser option uses the file **splaser.ini** to store configuration information. Here is a listing of the file **splaser.ini**:

```
COMSettings      COM3:9600,N,8,1
DiodeCurrent     100
PRF              500
DiodeOn          0
CutSpeed         100
```

17.8 Truly (Adtech) Data file Interface

Starting from user's defect location in Excel file.

User need to add 4 lines in the Excel file, then export it to become a *.ste file. (Test Site file)
The 4 lines are:

```
[TestSiteData]
Format=1
Scale=1
```

```
[DataStart]
.....
.....
[DataEnd]
```

where
Scale=1 ; if the unit for the defect location is micron
Scale=25.4 if the unit for the defect location is mil
Scale=1000 if the unit for the defect location is mm

Using Excel User need to Save As "Text (tab delimited)" file. The default extension for this type of file is *.txt to save as *.ste user need to enclose the file name with double quote e.g. "NGData01.ste"

Please see NGData01.ste as an example. The *.ste files are text files users can use the NotePad.exe program to view them.

The new version of the pmagic.exe (V5.23.3244) program can load the test site files.

After starting the software. Just switch to TestSite page on the Control Panel with the mouse pointed in the page press right mouse to pull down a menu
Select "Load Test Sites" to load the *.ste file.

After the test site are loaded, user can visit the site by selecting it and press the **Goto** button. For example select Loc#1 then press the **Goto** button will visit defect #1. After that pressing the Next button will move to the next defect location.

[TestSiteData]

Format=1

Scale=1

[DataStart]

12500	16940
12500	150740
12500	153680
12500	153820
12500	153960
12500	154940
12500	156830
12500	157390
33100	206780
77100	43700
77100	101980
77100	123980
77100	194020
297100	32900
297100	59730
297100	88870
297100	98270
297100	98690
297100	133360
297100	133500
297100	134270
297100	134550
297100	139240
297100	140990
297100	141480
297100	191920
320500	288320

[DataEnd]

17.9 ADTech Data File Interface

Introduction


The ADTech Data File Interface is a ProbeMagic™ extension DLL designed to read and interpret the data generated by an LCD panel inspection system. This interface greatly facilitates panel repair by utilizing the defect coordinate data generated by the inspection machine.


Software Installation

The interface is implemented in, **adtech.dll**, the DLL file located in the `..\ExtDLL` sub-directory. (Usually `C:\Pmagic\ExtDll`). The following steps load the interface into the ProbeMagic™ software:

6. In the command window, type **Config** <Enter>
7. Enter the password to launch the **System Configuration** dialog.
8. Select the **Options** page of the dialog.
9. In the **External Options** block, **Available Options** list, select **adtech.dll**.
10. Press the **Add ->** button.
11. Press the **OK** button.

User Interface

After the option is installed, the button  will appear in the button bar. Pressing the button launches the following dialog:



The screenshot shows a dialog box titled "ADTech Interface". It has a "File" section with the following fields and controls:

- Data File Name:** A text input field.
- Panel Number:** A dropdown menu showing "1" of "0".
- Comment:** A text input field.
- Location Number:** A dropdown menu showing "0" of "0".

At the bottom of the dialog, there are four buttons: "Prev", "First", "Next", and "Next Panel".



File : Click on File menu to load a data file.

Data File Name: Show the current Data file loaded.

Panel Number: Show the current panel inspection.

Location Number: Show the current location inspection.

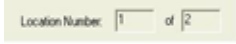
Prev. button : Move the stage back to the previous defect location.

First. button : Move the stage to the first defect location of the panel.

Next. Buton : Move the stage to the next defect location.

Next Panel. Button : Load the data for the next panel.

Panel repair operation steps

6. Place the panel onto the stage.
7. Set XY zero.
8. Load the corresponding ADTech data file by inserting the data 3.5 inch floppy disk in Drive A: of the computer. Select from the menu, **File, Load Data File.** (from **ADTech Interface** window)
9. Select the proper data file.
10. Press **Open.** Now the date file will loaded (see picture above.)
11. Click on the **First** button to located the first defect location for review or repair.
12. Then click on **Next** button to move to the next location (if the panel have more than one defect location.) 
13. After all of the defect sites have been review or repaired. Replace the next panel then click **Next Panel.** button to load the date for the next panel. (Redo step 6 through 8 until all panel have been inspected.)

Sample Data File Listing

Note: The line number and the comments are not part of the ADTech data file.

```
1: ???DATAFILE???
2: WORK_FILE=B:\WORKINGDATA.WRK
3: ID=00001 ;Panel Number
4: COMMENT= ;Comment
5: X=063050 ;X location of first defect
6: Y=242716 ;Y location of first defect
7: X=237469 ;X location of second defect
8: Y=246664 ;Y location of second defect
9: ID=00002
10: COMMENT=
11: X=237469
12: Y=246664
13: ID=00003
14: COMMENT=
15: X=237469
16: Y=246664
17: ID=00004
18: COMMENT=
19: X=295610
20: Y=205774
21: ID=00005
22: COMMENT=
23: X=237469
24: Y=246664
25: ID=00006
26: COMMENT=
27: X=237469
28: Y=246664
29: ID=00007
30: COMMENT=
31: X=237470
32: Y=163636
33: ID=00008 ;Panel Number
34: COMMENT= ;Comment
35: X=063050 ;X location of first defect
36: Y=076096 ;Y location of first defect
37: X=063050 ;X location of second defect
38: Y=103732 ;Y location of second defect
39: X=121190 ;X location of third defect
40: Y=129514 ;Y location of third defect
```

Appendix I ProbeMagic™ files

The installation program installs the following files into the computer's hard disk. The default base directory is C:\PMAGIC. The installation program also creates four sub-directories in the base directory: BIN, EXTDLL, EXAMPLE and USER.

<u>Name</u>	<u>Directory</u>	<u>Description</u>
PMAGIC.EXE	..\BIN	ProbeMagic executable file
PMAGIC.INI	..\BIN	ProbeMagic configuration file
PMHELP.HLP	..\BIN	Help file
CMDHELP.HLP	..\BIN	Help file 2.
NWLASER.DLL	..\BIN	Laser Cutter DLL
LASER.INI	..\BIN	Laser configuration
LASER.BMP	..\BIN	Button face for laser option
PANEL1.MPP	..\EXAMPLE	Examples map view program
TEST1.STE	..\EXAMPLE	Example test site program 1
TEST2.STE	..\EXAMPLE	Example test site program 2
DEMO1.SCR	..\EXAMPLE	Example script program 1
DEMO2.SCR	..\EXAMPLE	Example script program 2
DEMO3.SCR	..\EXAMPLE	Example script program 3
DEMO4.SCR	..\EXAMPLE	Example script program 4
DEMO5.SCR	..\EXAMPLE	Example script program 5
BUTTON1.BMP	..\USER	blank button face one
BUTTON2.BMP	..\USER	blank button face two

Appendix II Example Programs

The programs listed here are included in the ProbeMagic™ software package.

Panel1.mpp	-A simple map program
Test1.ste	-A simple test site program
Demo1.scr	-A very simple ProbeMagic program
Demo2.scr	-Variable usage and Loops
Demo3.scr	-Subroutine usage
Demo4.scr	-Keyboard and file I/O
Demo5.scr	-Decision making with IF THEN ELSE

All of the above programs are included in the ProbeMagic™ software package. They can be loaded into the ProbeMagic Editor (**File, Open, Script**), modified and then re-run (**File, Run Program**).

Wafer1.mpp

```
[Map0]
BoardSizeX=110000
BoardSizeY=110000
DieSizeX=10000
DieSizeY=10000
DieCornerX=4233.71
DieCornerY=1938.9
RefPosX=6428.29
RefPosY=-12600
GridOnFlag=1
SnapToGrid=1
FixDieSize=1
DisplayHomeDie=1
CDieColor=255
HDieColor=65535
DwellTime=1000
AssoFile=
```

```
[DieList]
4 5 0
3 5 0
2 5 0
1 5 0
1 6 0
2 6 0
3 6 0
4 6 0
5 6 0
6 6 0
2 8 0
1 7 0
3 3 0
[DieListEnd]
```

Test1.ste

```
[TestSiteProgram]

Dwell_Time          = 1000
Associated_Script_File = NONE
TestSitesLocalToDie = 0

[TestSites]
Loc#1      0.000    0.000
Loc#2     10000.000 -2000.000
Loc#3     -5000.000 10000.000
Loc#A      1000.000 -2000.000
Loc#B     -8000.000  1000.000
[End]
```

Demo1.scr

```
; This file is a very simple ProbeMagic™ script file
; just to get you started.
```

```
BEGIN                                ; START OF ProbeMagic™ PROGRAM
DISPN,"This is a simple ProbeMagic program" ; DISPLAYS A MESSAGE
SCALE,XY,25.4                        ; SETS SCALE TO mil.
                                      ; THE BASE SCALE IS um
MOVE,100,100                          ; MOVES TO LOCATION 100,100
MR,-100,-100                          ; MOVES RELATIVE
DISPN,"It is easy to program the ProbeMagic"
END                                    ;END OF SCRIPT PROGRAM
```

Demo2.scr

; This ProbeMagic™ script program demonstrates:

- ; 1. iteration - LOOP and LEND command.
- ; 2. user variables and maths - VAR ,LET etc

```
VAR,%A,%B,%C,V1,V2,A,B      ;DECLARES USER VARIABLES
BEGIN
LET, %A =100                 ;ASSIGNS VALUE TO USER VARIABLE
LET,%B=200
LET,%C=(%A>%B) OR (%A>10)
LET,V1=( %A - %B) * 2 /40    ;MATH EXPRESSIONS
LET,V2=100 ^ 2               ;EXPONENTIAL
DISPN,"V1=",V1," V2=",V2," C=",%C
DELAY                        ;WAITS FOR USER KEY STROKE
LET,A=2
LOOP,A                        ;ITERATIONS BY USING LOOP
                              ;COMMAND
LOOP,3                        ;LOOP COMMAND CAN BE NESTED
LET,B=B+1
DISPN,"B= ",B
DELAY,500                    ;WAITS 500ms
LEND                          ;ENDS LOOP
LEND                          ;ENDS LOOP
END
```


Demo3.scr

This ProbeMagic script program demonstrates the subroutine function.
; e.g. SUB, CALL, RET

```
SUB,ONE                                ;SUBROUTINE ONE
DISPN,"This is subroutine ONE"
RET                                     ;RETURN

SUB,TWO
DISPN,"This is subroutine TWO"
CALL,FUN                               ;CALLS ANOTHER SUBROUTINE
RET

SUB,FUN
DISPN,"ProbeMagic programing is fun !"
RET

BEGIN
CALL,ONE
CALL,TWO
DELAY                                  ;WAITS FOR ENTER KEY PRESS
LOOP,3                                  ;CALLS FUN 3 TIMES
CALL,FUN
LEND
END
```

Demo4.scr

;This ProbeMagic™ script program demonstrates the input/output capabilities
;of the software.

```
VAR,A,B                               ;DECLARES VARIABLES
BEGIN
DISPN," Keyboard I/O"
DISPN," The INKEY and INPUT commands"
INKEY,A
let,b=10
INPUT,B
DISPN,"A= ",a," B= ",b
DISPN," File and Printer I/O"
DISPN," The READ and WRITE commands"
OPEN,DEMO.DAT,IN
READ,A,B                               ;READS VALUES FROM FILE
SETAXIS,1
ACCEL,XY,B                             ;SETS ACCELERATION
SPEED,XY,A                             ;SETS SPEED
CLOSE,IN                               ;CLOSES FILE
OPEN,DEMO.OUT,OUT                      ;OPENS OUTPUT FILE
WRITE,"THIS IS THE OUTPUT FILE "       ;WRITES TO FILE
WRITE,"A= ",A," B=",B
CLOSE,OUT                              ;CLOSES OUTPUT FILE
DISPN,"If you want to write to the "
DISPN,"printer simply open a file "
DISPN,"named PRN. i.e. OPEN,PRN.OUT"
END
```

Demo.Dat (The data file read by demo4.scr)

```
; This is a ProbeMagic™ data file
11           ; first data
22,33       ; multiple data can be written in one line
```

Demo5.scr

;This ProbeMagic™ program demonstrates its decision making capacity.

```
SUB,B=6
DISPN,"B IS EQUAL TO 6"
RET

VAR,A,B
BEGIN
INKEY,A
IF A>5 THEN goto ,1 ELSE goto, 2      ;IF_THEN_ELSE WITH GOTO
:1
DISPN,"THE NUMBER IS > 5"
GOTO ,3
:2
DISPN,"THIS NUMBER IS <=5"
:3
INKEY,B
IF B=6 THEN CALL,B=6                  ;IF_THEN WITH CALL
IF ((A +B)>10) OR (A>B) THEN DISP,N,"TRUE" ELSE DISP,N,"FALSE"
END
```

Appendix III A Sample Listing of PMAGIC.INI

```
[MainFrame]
Left=0
Top=0
Right=700
Bottom=600
IsZoomed=1
IsColor=1
CurrLens=1

[Config]
ProbeMagic=Version 4.00
SoftDate=Feb-15-98
Program=PMAGIC
Privilege=3
UserLog=0
StartupFile=NONE
GPIB_Interface=0
GPIB_Master=0
GPIB_SlaveAddress=7
ReportGPIB=0
MotorizedTheta=1
ThetaStep=0.2
ThetaJogSpeed=0.2
MT405Port=2
Lens1=5x
Lens2=10x
Lens3=20x
Lens4=50x
Lens5=100x
FocusDelay=5000
FocusScale=1
FocusMotorHold=0
FocusPos=0
FocusStep1=100
FocusStep2=50
FocusStep3=25
FocusStep4=10
FocusStep5=5
EdgeSensorState=0
EdgeSensor1Mask=
EdgeSensor2Mask=
EdgeSensor3Mask=
EdgeSensor4Mask=
ProbeMotorMask=
SeparatePos=100
OverDrive=0
NoPlatenZ=0
TrackEnable=0

[Control]
JogSpeed1=10000
```

JogSpeed2=1000
JogSpeed3=100
JogSpdIdx=0
ZJogSpeed=100
XJogStep=10000
YJogStep=10000
ZJogStep=100

[UserButton]
Btn1=
Btn2=
Btn3=
Btn4=

[Channel]
Theta=0
Skew=0
AlignDx=100000
AlignDy=0
AlignPx=0
AlignPy=0
AlignRx=0
AlignRy=0
AlignMaxTry=2
AlignAngRange=0.1
Cali0X=0
Cali0Y=0
Cali1X=100000
Cali1Y=0
Cali2X=0
Cali2Y=100000
CaliPX=0
CaliPY=0
XHomeOff=0
YHomeOff=0
SwapXY=0

[Axis1]
Pos=0
Speed=1000
Accel=10000
Dir=1
Scale=1000
MotorRel=1
EncRel=0
ErrComp=1
EdgeSensor=0

[Axis2]
Pos=0
Speed=1000
Accel=10000
Dir=1
Scale=1000
MotorRel=1

EncRel=0
ErrComp=1
EdgeSensor=0

[Axis3]
Pos=0
Speed=1000
Accel=10000
Dir=1
Scale=1000
MotorRel=1
EncRel=0
ErrComp=1
EdgeSensor=4

[EdgeSensor]
MotorMask=15
SMask1=1
SMask2=2
SMask3=4
SMask4=8
OverDrive=0

[Axis4]
Pos=0
Speed=1000
Accel=10000
Dir=1
Scale=1000
MotorRel=1
EncRel=0
ErrComp=1
EdgeSensor=0

[CmdFrame]
Left=652
Top=0
Right=960
Bottom=130

[MapFrame]
Left=652
Top=131
Right=960
Bottom=420
ShowMap=0

[ControlRect]
Left=660
Top=420
Right=960
Bottom=680

[DispNRect]
Left=0

Top=510
Right=650
Bottom=640

[COMPorts]
COM1Enable=0
COM1Setting=COM1:9600,N,8,1
COM2Enable=0
COM2Setting=COM2:9600,N,8,1
COM3Enable=0
COM3Setting=COM3:9600,N,8,1
COM4Enable=0
COM4Setting=COM4:9600,N,8,1

[ExtDLL]
DLL#1=

[VisionAlign]
Enabled=0
LALeft=150
LATop=150
LARight=490
LABottom=330
RALeft=150
RATop=150
RARight=490
RABottom=330
LSCx=100
LSCy=100
RSCx=100
RSCy=100
CameraUsed=0
PassScore=750
Retry=1

[AutoFocus0]
Left=220
Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus1]
Left=220
Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus2]
Left=220

Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus3]
Left=220
Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus4]
Left=220
Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus5]
Left=220
Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus6]
Left=220
Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus7]
Left=220
Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus8]
Left=220
Top=140

Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[AutoFocus9]
Left=220
Top=140
Right=420
Bottom=340
StepSize=100
StepRange=2
PassScore=30000

[PatRec0]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec1]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec2]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec3]
Left=120
Top=80
Right=520

Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec4]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec5]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec6]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec7]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec8]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[PatRec9]
Left=120
Top=80
Right=520
Bottom=400
TmpLeft=270
TmpTop=190
TmpRight=370
TmpBottom=290
PassScore=700
ActionType=0

[VideoFrame]
Left=0
Top=0
Right=650
Bottom=510
IsZoomed=0

Appendix IV A Sample Listing of LVIDEO.INI

```
[LiveVideo]
Board=METEOR
LineWidth=1
StatusFlag=6
XhairX=0.5
XhairY=0.5
XhairLen=0.1
ScaleX=60.8223
ScaleY=65.7366
ScaleRef=0
CScale0X=38926.3
CScale0Y=31619.3
CScale0Ref=0
CScale0XOff=0
CScale0YOff=0
CScale0Theta=0
CScale1X=18000
CScale1Y=14150
CScale1Ref=33
CScale1XOff=730.951
CScale1YOff=57.9988
CScale1Theta=0
CScale2X=11000
CScale2Y=8700
CScale2Ref=66
CScale2XOff=730.951
CScale2YOff=57.9988
CScale2Theta=0
CScale3X=10000
CScale3Y=10000
CScale3Ref=99
CScale3XOff=0
CScale3YOff=0
CScale3Theta=0
RoiWidth=640
RoiHeight=480
RoiXOffset=0
RoiYOffset=0
```