PMAC-NC Designer Manual
Copyright Information
© 2007 Delta Tau Data Systems, Inc. All rights reserved.
This document is furnished for the customers of Delta Tau Data Systems, Inc. Other uses are unauthorized without written permission of Delta Tau Data Systems, Inc. Information contained in this manual may be updated from time-to-time due to product improvements, etc., and may not conform in every respect to former issues.
To report errors or inconsistencies, call or email:

Delta Tau Data Systems, Inc. Technical Support
Phone: (818) 717-5656
Fax: (818) 998-7807
Email: support@deltatau.com
Website: http://www.deltatau.com

Operating Conditions
All Delta Tau Data Systems, Inc. motion controller products, accessories, and amplifiers contain static sensitive components that can be damaged by incorrect handling. When installing or handling Delta Tau Data Systems, Inc. products, avoid contact with highly insulated materials. Only qualified personnel should be allowed to handle this equipment.
In the case of industrial applications, we expect our products to be protected from hazardous or conductive materials and/or environments that could cause harm to the controller by damaging components or causing electrical shorts. When our products are used in an industrial environment, install them into an industrial electrical cabinet or industrial PC to protect them from excessive or corrosive moisture, abnormal ambient temperatures, and conductive materials. If Delta Tau Data Systems, Inc. products are exposed to hazardous or conductive materials and/or environments, we cannot guarantee their operation.
<table>
<thead>
<tr>
<th>REV.</th>
<th>DESCRIPTION</th>
<th>DATE</th>
<th>CHG</th>
<th>APPVD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CREATION OF MANUAL</td>
<td>12/12/06</td>
<td>CP</td>
<td>D. GHAZARIAN</td>
</tr>
<tr>
<td>2</td>
<td>FULL RELEASE OF MANUAL</td>
<td>09/28/07</td>
<td>CP</td>
<td>V. BUROKAS</td>
</tr>
</tbody>
</table>
# Table of Contents

## INTRODUCTION

## SECTION ONE

- frmMain
- frmGMPosDisp
  - Absolute and machine position displays
- Spindle Status
- Feedrate Status
- Active tool Status
- Active M Codes
- Active G Codes
- frmStatusPane
- frmNCStatus
- frmPosAll
- frmMainMenu

## SECTION 2: SUB-MENUS

- POS(F1) menu item
- PROG(F2) menu item
- OFFSET(F3) menu item
- TOOLS(F4) menu item
- EDITOR(F5) menu item
- DIAG(F6) menu item
- Parametric Variables window (frmParameteric)
- Terminal Window (frmTerminal)
- Real Time Plot window (frmRealTimePlot)
- Data Pages Window (frmDataPages)
- Language Setup (frmLanguageSetup)
- MSGS(F7) menu item
  - Errors message Window (frmErrorList)
- OPER(F8) menu item
  - Operator Window (frmOperator)
- ALT-F Keys

## SECTION 3: SAMPLES
INTRODUCTION

The PMAC-NC Designer Manual is designed to assist developers who wish to modify and add functions and screens to the NC 5.0 program by using the PMAC-NC Designer.

This manual is divided in three sections.

- **Section One**: This section explains the NC5.0 main screen and its overall structure, as well as the functions used by NC 5.0 project to design the NC 5.0.

- **Section Two**: This section explains the NC5.0 sub-menus and its overall structure, as well as the functions used by NC 5.0 project to design the NC 5.0.

- **Section Three**: This section demonstrates how to create a sample user form and add it to your NC 5.0 project.
SECTION ONE

The NC 5.0 main page consists of six individual forms. Each form is arranged next to the other form to compose the NC 5.0 project.

The following is the main screen of NC 5.0, with each of the six sections described below:

Here are the forms that make up the NC5.0 Main screen.

1) frmMain
2) frmGMPosDisp
3) frmStatusPane
4) frmNCStatus
5) frmPosAl
6) frmMainMenu

By design, sections 1 through 4 are the fixed areas of NC 5.0 project and the sections 5 and 6 are the changeable sections. Section 5 is the menu of NC 5.0 project. These sections change whenever a main menu item is clicked and is replaced by a submenu. Section 6 is reserved to display any other screen that in part of NC 5.0 project and also for the users to add their own custom screen and display it in this section. Users can also replace any other screen in the fixed section of NC 5.0 with their own custom screen.

First we will start by describing each form in the main screen and their components. Each form in NC 5.0 consists of many ActiveX controls (components). Some of these controls are built into the PHMI, but a few are third-party components. The description of each form will include the functions being called by
each of these components. Then we will follow the main menu on NC 5.0 and describe any screen that gets displayed by the submenus.

**frmMain**

**Form main** consists of the NC program display window and MDI editor. When NC is running in Manual or Auto mode, the NC program display window will be maximized.

To display the NC program, use the following command:

**Devices.PmacNC1.GetPEWCmd**

GetPEWCmd will automatically load the NC program into the NC display window.

When NC is running in MDI mode, the NC display window will be resized and the MDI Edit window will be visible.

The **Read NC** program button will load the last saved MDI program in the MDI editor window. Function Used: **GetBufferText**

The **Write NC** Program button will load the program into the NC buffer. Function Used: **SetBufferText**
frmGMPosDisp

Form GMPosDisp displays the Absolute and Machine positions, Spindle status, Feedrate status, Active tool status, Active M code status, and active G code status.

### Absolute and Machine position displays
These position displays use NC position display ActiveX control to display their respective positions. The NC position display activeX control encapsulates all the position display functionality within itself. Users can easily drag and drop this control into any form and, from its property, select the desired position display type.

### Spindle Status
Spindle status shows the status of the spindle. The following are the functions used to display the spindle status.

<table>
<thead>
<tr>
<th>Spindle Status: OFF</th>
<th>GetSpindleStatus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Speed: 6000</td>
<td>GetSpindleMaxRPM</td>
</tr>
<tr>
<td>Cmd Speed: 150.0</td>
<td>GetSpindleCmdRPM</td>
</tr>
<tr>
<td>Act Speed: 0.0</td>
<td>GetSpindleActRPM</td>
</tr>
<tr>
<td>% Override: 100</td>
<td>GetSpindleOverride(0)</td>
</tr>
<tr>
<td>CSS Mode: OFF</td>
<td>GetSpindleCSSMode</td>
</tr>
</tbody>
</table>
Feedrate Status

Feedrate status displays the status of the feedrate. The following are the functions used to display the feedrate status.

<table>
<thead>
<tr>
<th>Feedrate</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Feed</td>
<td>200.0</td>
</tr>
<tr>
<td>Cmd Feed</td>
<td>200.0</td>
</tr>
<tr>
<td>Act Feed</td>
<td>0.0</td>
</tr>
<tr>
<td>% Override</td>
<td>100</td>
</tr>
<tr>
<td>Mode</td>
<td>FPM</td>
</tr>
<tr>
<td>% Rapid</td>
<td>100</td>
</tr>
</tbody>
</table>

- GetMaxFeedrates
- GetFeedrate
- GetVectorVelocity(0)
- GetFeedrateOverride(0)
- GetTimebaseMode(0)
- GetRapidOverride

Active tool Status

Active tool status displays the active tools and the offsets. The following are the functions used to display the active tools.

<table>
<thead>
<tr>
<th>Active Tool</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tool No</td>
<td>T0</td>
</tr>
<tr>
<td>Offset</td>
<td>H00 D00</td>
</tr>
</tbody>
</table>

- GetToolInfo(0)
- GetCompOffset(0)
- GetToolOffset(0)
Active M Codes
The Active M Codes window displays the current list of active M codes. The following are the functions used to display the active M codes.

Active G Codes
The Active G Codes window displays the current list of active G codes. The following are the functions used to display the active G codes.
**frmStatusPane**

The *Form statuspane* displays the operator panel status in manual mode, program status in auto mode and error messages. The status of operator mode in manual mode and the program status in auto mode are queried through seven DQuery objects. Each query object queries for a specific function. The queries are enabled and disabled based on the Machine status (Manual, Auto or MDI modes).

The *Error Message* display window lists all errors and warnings defined in errors.Dat file. The errors are queried through a Dquery control and the result is displayed in the error display window. The Error display window can display 4 types of messages:

1) If the message is from NC, it is level 0 and displayed in **Lime** color
2) If the message is a Warning from NC, it is level 1 and displayed in **Yellow** color
3) If the message is a Stop message from NC, it is level 2 and displayed in **Fuchsia** color
4) If the message is an Error from NC, it is level 3 and displayed in **Red** color

The following are the functions used to display operator panel status in Manual mode:
The following are the functions used to display the program status in Auto and MDI modes:

- `GetProgramMode`
- `GetInPosition`
- `GetBufferOpen`
- `GetBufferRemaining(0)`
- `GetSingleBlock`
- `GetOptionalStop`
- `GetBlockDelete`

### frmNCStatus

**Form NCStatus** displays the current loaded program in the NC buffer, the Program status and the current label. The following are the functions used:

- `GetProgramPath(0,0)`
- `GetProgramRepeatStatus`
- `GetProgramLineStatus`
- `GetCurrentLabel`
frmPosAll
Form PosAll displays NC Axis positions. It displays Relative, Machine, Commanded, and following Errors. As described in frmGMPosDisp, all the position display windows are part of the NC position display window ActiveX control and encapsulate all information needed to display the proper position.

<table>
<thead>
<tr>
<th>Relative</th>
<th>Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>X 4.9693</td>
<td>X 4.9693</td>
</tr>
<tr>
<td>Y 4.9693</td>
<td>Y 4.9693</td>
</tr>
<tr>
<td>Z 5.9693</td>
<td>Z 5.9693</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Commanded</th>
<th>Following Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>X 4.9693</td>
<td>X -0.0000</td>
</tr>
<tr>
<td>Y 4.9693</td>
<td>Y 0.0000</td>
</tr>
<tr>
<td>Z 5.9693</td>
<td>Z -0.0000</td>
</tr>
</tbody>
</table>

frmMainMenu
Form MainMenu controls the NC project menu structure. Each menu item in the Main menu is part of a Form group (please refer to the PHMI Manual for more information). The Form Groups functionality is to show or hide another form within its group. To display or hide any menu, we have created the SubMenuGroup as part of the form groups. The following forms belong to this group:

1) frmDiagnosticMenu
2) frmEditorMenu
3) frmEditorMenu2 (second level menu of the editor)
4) frmErrorsMenu
5) frmMainMenu
6) frmOperatorMenu
7) frmPositionMenu
8) frmProgramMenu
9) frmtoolMenu
10) frmworkOffsetMenu
For example, if the Editor menu item is clicked:

(function Used: \textit{Forms.SubMenuGroupfrmEditorMenu.Show})

it displays \textit{frmEditorMenu} and automatically hides \textit{frmMainMenu}. In general, this is a very easy and convenient way to display and hide forms in PHMI. Also, each menu item is attached to an F-key displayed on the top of each menu item.

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS</td>
<td>PROG</td>
<td>OFFSET</td>
<td>TOOLS</td>
<td>EDITOR</td>
<td>DIAG</td>
<td>MSGS</td>
<td>OPER</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following are the commands used to display each sub-menu by clicking or pressing an F-key:

1) \textit{Forms.SubMenuGroupfrmPositionMenu.Show}
2) \textit{Forms.SubMenuGroupfrmProgramMenu.Show}
3) \textit{Forms.SubMenuGroupfrmworkOffsetMenu.Show}
4) \textit{Forms.SubMenuGroupfrmtoolMenu.Show}
5) \textit{Forms.SubMenuGroupfrmEditorMenu.Show}
6) \textit{Forms.SubMenuGroupfrmDiagnosticMenu.Show}
7) \textit{Forms.SubMenuGroupfrmErrorsmenu.Show}
8) \textit{Forms.SubMenuGroupfrmOperatorMenu.Show}
SECTION 2: SUB-MENUS

This section covers menu options, sub-menus, and the screens that accompany each specific menu item. We will follow the main menu buttons and explain what each button does and which screen it will display. Each menu item has a sub-menu and at least one screen associated with it.

Each sub-menu form has its own Forms Group. For example, sub-menu POS has the Forms Group named PosGroup and has frmPOSAll included in this group. Every sub-menu group except DiagGroup has only one form. The sub-menu Forms Groups is used to display the forms that start up at default by clicking on the sub-menu. To demonstrate how the Forms Group and default page work, here is an example.

The following is the sequence of events that displays the POS sub-menu item and starts the frmPosAll:

1. When the user presses the POS sub menu item, the command
   
   `Forms.SubMenuGroup.frmPositionMenu.Show`

   is sent to PHMI.

2. PHMI displays the SUB menu POS based on the command that was sent.

   *At the same time, to show the frmPosAll we have to do the following.*

3. In the run mode properties of frmPositionMenu, we select Form Commands Tab (right-click on the form and select Run Mode Properties, then select the Form Commands tab).

4. Make sure the “Execute command when the form shows” is checked. This ensures that any command entered there will be executed when the Position sub-menu is shown.

5. Enter the following commands:

   ```
   Forms.PosGroup.ShowLast
   DataTables.GlobalVars.CurrentMenu = 1
   VBA.frmMain.DoMDIFocus
   ```
6. **Forms.PosGroup.ShowLast** is going to show the last form that was displayed in the PosGroup. Since we have only one form defined in this Forms Group(frmPosAll), it means that it will always show frmPosAll. If there were more forms as part of this group, then the last form shown would have been displayed.

7. **DataTables.GlobalVars.CurrentMenu = 1** sets the current menu level. This is used to command F-keys to show the appropriate sub-menus based on the menu level.

8. **VBA.frmMain.DoMDIFocus** command is always present to force the focus to MDI mode whenever the MDI mode is selected.

9. As a result, whenever the Position sub-menu is displayed, the above commands will be executed and frmPosAll will be displayed each time the Position sub-menu is selected.

Please note that this rule applies to every sub-menu form and its dependent forms. Therefore in the following descriptions of each sub-menu item, this manual will not go into detail for each one of them.

**POS(F1) menu item**

By clicking on the menu item POS or pressing F1, the Position sub-menu form (frmPositionMenu) will be displayed. Also, the Form Position All (frmPosAll) will be shown. Since frmPosAll is part of the main NC screen and we have already described it, this section will describe only the functions of the Position sub-menu. Each sub-menu’s first button is the **BACK (F1)** button. By clicking on the Back button, the user will return to the previous menu which, in all cases except in the editor’s second level menu, will be the main menu. The clear origin, CL ORG, button will clear all the origins and Origin All button, ORG ALL, will origin all the positions. The following are the functions used:

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;BACK</td>
<td>CL ORG</td>
<td>ORG ALL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Forms.SubMenuGroup.frmMainMenu.Show**
- **SetOrigin(1)**
- **SetOrigin(0)**
**PROG(F2) menu item**

By clicking on menu item PROG or pressing F2, the Program sub-menu form (frmProgramMenu) will be shown. Also, the Form Program status (frmProgStat) will be shown. Form ProgStat displays all the machine specific time information, Parts information and the current date and time. The following are the functions used:

- GetOperatingTime
- GetRunningTime
- GetPartsRequired
- GetPartsCount
- GetPartsTotal
- GetDate
- GetCycleCuttingTime
- GetCycleTime
- GetPartsRequired
- GetPartsCount
- GetPartsTotal
- GetTime
The PROG sub menu has the following buttons. The **BACK** button will go back to the main menu. The **LOAD** button will open up a file select dialog box so the user can select the NC file to be opened. The **REWIND** button will rewind the currently loaded NC program to the beginning. The **SEARCH** button will open up a search dialog box for searching the entire text of the currently loaded NC program. **GOTO LINE** will open up an input box for going to a specific line within the currently loaded NC program. **Reset Parts Count** will reset the parts count, **Reset Parts Total** will reset the parts total and the **Reset Cycle Time** will reset the cycle time. The following are the functions used:

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;BACK</td>
<td>LOAD</td>
<td>REWIND</td>
<td>SEARCH</td>
<td>GOTO LINE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Diagram showing the buttons and functions](image_url)

**Please enter the text to be searched**

![Search dialog box](image_url)

**Please enter the line number**

![Search line dialog box](image_url)
OFFSET (F3) menu item
By clicking on menu item OFFSET or pressing F3, the Program sub menu-form (frmWorkOffsetMenu) will be displayed. Also, the Form Work Offset (frmWrkOffset) will be shown. Form Work Offset displays all the work offsets of the Machine for the active axis. The GetWorkOffsetArray Function is used to get all the work offsets.

The OFFSET sub menu has the following buttons. The BACK button will go back to the main menu. The SETX, SETY, SETZ, SETA, SETB, and SETC buttons set the work offset for the desired axis. The ALL button will set the work offset for all the axes. The SetOffsetVal function is used to set the offsets for each axis.

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;BACK</td>
<td>SET X</td>
<td>SET Y</td>
<td>SET Z</td>
<td>SET A</td>
<td>SET B</td>
<td>SET C</td>
<td>ALL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TOOLS (F4) menu item
By clicking on the menu item TOOLS or pressing F4, the Program sub-menu form (frmToolMenu) will be displayed. Also, the Form Tool Offset (frmToolOffset) will be shown. The Form Tool Offset displays all the Tool offsets of the Machine. The GetToolOffsetArray Function is used to get all the Tool offsets.

The TOOLS sub-menu has the following buttons. The BACK button will go back to the main menu. The SETX and SETZ buttons set the Tool offset for the desired axis. The AutoSetToolOffset function is used to set the offsets for each axis.

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;BACK</td>
<td>SET X</td>
<td>SET Z</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

EDITOR (F5) menu item
By clicking on the menu item EDITOR or pressing F5, the Editor sub-menu form (frmEditorMenu) will be displayed. Also, the Form Editor(frmEditor) will be shown. The Editor is used for creating and modifying NC programs. The Editor menu, unlike any other menu structure, has two levels of menus. MORE button will take the user to the second level of editor menu, which will be explained after the first level menu.

Here is a description of the functions in the editor’s first level menu. The BACK button will go back to the main menu. The Edit Curr. Program will unload the currently loaded program from the NC buffer, if once exists, and will load it into the editor. The Load Curr. Program will load the text in the editor screen, if one exists, into the NC buffer. SAVE will save the program in the editor. FIND will let the user find a specific text within the editor. The edit commands REPLACE, CUT, COPY, PASTE, UNDO, REDO are self-explanatory.

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;&lt;BACK</td>
<td>Edit Curr. Program</td>
<td>Load Curr. Program</td>
<td>SAVE</td>
<td>FIND</td>
<td>REPLACE</td>
<td>CUT</td>
<td>COPY</td>
<td>PASTE</td>
<td>UNDO</td>
<td>REDO</td>
<td>MORE &gt;&gt;</td>
</tr>
</tbody>
</table>
The **MORE** menu option will display the second level on the editor Menu:

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=BACK</td>
<td>NEW</td>
<td>OPEN</td>
<td>SAVE AS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The **Back** button returns the user to the Editor menu level 1. **New, Open, and SaveAs** buttons are self-explanatory.

**DIAG(F6) menu item**

By clicking on menu item DIAG or pressing F6, the **Diagnostic** sub-menu form (frmDiagMenu) will be displayed. The Diagnostic option consists of 5 different forms (utilities) that will be displayed by clicking on the appropriate menu option. The Diagnostic option consists of the following forms:

1. **Parametric Variables (frmParametric)**
2. **Terminal Window (frmTerminal)**
3. **Real Time Plot (frmRealTimePlot)**
4. **Data Pages (frmDataPages)**
5. **Language Setup (frmLanguageSetup)**

All the forms within the Diagnostic option belong to the **DiagGroup** forms group. To display each from within the diagnostic group, the command `Forms.DiagGroup.FormsName.Show` is used. For example, to show the form `parametric` the command `Forms.DiagGroup.frmParametric.Show` is used.

<table>
<thead>
<tr>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
<th>F9</th>
<th>F10</th>
<th>F11</th>
<th>F12</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;=Back</td>
<td>PRM VAR</td>
<td>TERMINAL</td>
<td>Brick I/O</td>
<td>PLOT</td>
<td>Data Pages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following is the description of each form within the diagnostic group and the functions used:
Parametric Variables window (frmParameteric)
The Parametric window allows users to program the parametric variables. To read the parametric variables, the function `GetParamValues` is used and the set a value `SetParamValue` is used.

<table>
<thead>
<tr>
<th>LOCAL VARIABLES #1-33</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 A</td>
<td>0.000000</td>
</tr>
<tr>
<td>#2 B</td>
<td>0.000000</td>
</tr>
<tr>
<td>#3 C</td>
<td>0.000000</td>
</tr>
<tr>
<td>#4 I</td>
<td>0.000000</td>
</tr>
<tr>
<td>#5 J</td>
<td>0.000000</td>
</tr>
<tr>
<td>#6 K</td>
<td>0.000000</td>
</tr>
<tr>
<td>#7 D</td>
<td>0.000000</td>
</tr>
<tr>
<td>#8 E</td>
<td>0.000000</td>
</tr>
<tr>
<td>#9 F</td>
<td>0.000000</td>
</tr>
<tr>
<td>#10</td>
<td>0.000000</td>
</tr>
<tr>
<td>#11 H</td>
<td>0.000000</td>
</tr>
<tr>
<td>#12</td>
<td>0.000000</td>
</tr>
<tr>
<td>#13 M</td>
<td>0.000000</td>
</tr>
<tr>
<td>#14</td>
<td>0.000000</td>
</tr>
<tr>
<td>#15</td>
<td>0.000000</td>
</tr>
<tr>
<td>#16</td>
<td>0.000000</td>
</tr>
<tr>
<td>#17 Q</td>
<td>0.000000</td>
</tr>
<tr>
<td>#18 R</td>
<td>0.000000</td>
</tr>
<tr>
<td>#19 S</td>
<td>0.000000</td>
</tr>
<tr>
<td>#20 T</td>
<td>0.000000</td>
</tr>
<tr>
<td>#21 U</td>
<td>0.000000</td>
</tr>
<tr>
<td>#22 V</td>
<td>0.000000</td>
</tr>
</tbody>
</table>
Terminal Window (frmTerminal)
The Terminal window allows full access to the PMAC. Users can type any valid PMAC command into the terminal window. This Terminal window is a DTextTerm control, which is one of the PHMI’s built-in controls. The Terminal window is protected by a password to control direct access to the PMAC. Any time a user enters the Terminal window, the following password screen is displayed.

The password to access the terminal is **delta**.

To reset the password for a screen, please refer to the PHMI Design Manual.
Real Time Plot window (frmRealTimePlot)
The Real Time Plot window plots the real time data gathered by PcommServer. The built-in DPlot ActiveX control in PHMI is capable of intercepting the real time data and displaying it. To display the real time data, the __DATASHOW command is being used in the traces of the DPlot control.

The DataStart function will start the realtime data and the DataStop function will stop the real time plot. Top Plot Settings and Bottom Plot settings will display the set up dialog in which the users can setup the data to plot and also the color, label, scales, etc.
**Data Pages Window (frmDataPages)**

Data Pages is a basically a watch window in which users can enter variables to monitor. Data Pages is an ActiveX control which has all its functionality encapsulated within the ActiveX. Data Pages allow users to have up to 250 pages of data, with each page containing up to 250 variables. To enter a variable into the Data Pages, Click on the **Insert** key of the keyboard and the following window will popup.

**Watch Parameter**: the parameter that the user wants to monitor.

**Variable Label Name**: the arbitrary name that the user can choose to assign to any parameter.

**Set Current value**: allows the users to change the value of the watch parameter.

**Page label name**: the name of the current page that the user wants to assign.

**Show variable check box**: enables or disables the display of the variable.
After entering a new parameter into the data pages, users can double-click on any parameter and access their entered data through the same window. Users may also change the value of the entered parameter.

All the data will be stored in a simple text file. The location of this file will be determined through the registry. The following key, located at:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PMAC\Device0\Nc0\System\DiagPagesFilePath

has the information where the file is located. The format of the file is the following:

[DIAG_SETUP]
DATA_PRECISION:5

[0;sample Page]
M250;Machine Output1;1
p21;Brake On;1
p22;Brake Off;1
p1;test;1

[DIAG_SETUP] key word indicates the parameter within its group is a setup parameter. The only parameter within its group is called DATA_PRECISION. This parameter sets the data precision for every data that is going to be monitored.

[0,Sample Page] key word is the actual page that has been created by Data Pages. The first parameter 0 is the page number (they are indexed from 0). So the next page number will be 1. The Sample Page is the page name that the user has entered for this page. All the contents within this Page are the data that the user has entered and wishes to be monitored. The format of each data is the following:

Watch Parameter;Variable Label Name; Show variable check box

Each item in the above format is distinguished by a semicolon. For example, the first item in the Sample Page is M250;Machine Output1;1. Users can enter up to 250 items per each page and up to 250 pages. Users can only create one page through the Data Pages visual interface. To create extra pages, users must modify the file that has the actual saved data. For example, to add another page to the Data Pages, look at the following:
[DIAG_SETUP]
DATA_PRECISION:5

[0;sample Page]
M250;Machine Output1;1
p21;Brake On;1
p22;Brake Off;1
p1;test;1

[1;The Second Page]
P100;Sample Variabe;1
M500;Input 1;1

Users can click on **Reload Data Pages** to see the added page. The **Select Data Pages** list box has all the pages that users have created, and these can be selected here. To delete an item while viewing the variable, press the **Delete** key of your keyboard to remove the item. Also, **page up** and **page down** will
move the highlighted item 10 parameters up or down. Similarly, the **scroll up** or **down** buttons will scroll the highlighted variable up or down.

**Language Setup (frmLanguageSetup)**

The **Language Setup** window allows users to select the language in which they would like to view NC5.0. Users will always have a choice of between the running NC in English and the language in their version of Windows. For example, a Japanese Windows OS can run NC in either English or Japanese. The Language Setup screen automatically detects the version of the installed OS and disables the languages that cannot be applied. After selecting the desired language, users must apply the changes which update the registry location and records the information where the NC language file resides. The registry location is:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\PMAC\Device0\Nc0\System\NCLangDefaultPath

The default file name for NC language is called **NCLang English.txt**. The same file in Japanese would be **NCLang Japanese.txt**. After applying the changes, the user must restart the NC5.0 program in order to see the changes.

Language integration in the PMAC Is handled through an ActiveX called NcLanguageX. This ActiveX control reads the context of a file which has all the captions of objects of NC program (the file name as described above is **NCLang English.txt**). The format of the text in this file is the following:
<The form name>
[The object name within the form : Caption of the object]

For example, in the form Diagnostic Menu (frmDiagnosticMenu) there are six buttons. The format of the text represents the captions of those buttonse:

`<frmDiagnosticMenu>
[btnBack : <<Back]`
`[btnPrmVar : PRM /r VAR]`
`[btnTerminal : TERMINAL]`
`[btnPlot : PLOT]`
`[btnDataPages : DATA /r PAGES]`
`[btnLangSetup : LANG. /r SETUP]`

Note that the characters /r are present on some of the lines. This is a carriage return sign. When text is applied to the actual object, a carriage return signals that it will be displayed in two lines rather than one.

NCLanguageX ActiveX control reads all the information within the above text file and stores it in internal memory for retrieval by NC5.0. NCLanguageX control has two functions:

1) **LoadLanguageText**(NC Language File Path): This function loads the contents of NC language text file into NCLanguageX internal memory

2) **GetObjectCaption**("Form name", "Object name"): This function will return the actual caption of an object, based on the form name and the object name.

To use the NCLanguageX Control, users must drag and drop this control onto the desired form. Then in the form initialize they should do the following (this is an example from the Form Diagnostic menu):

```vbnet
Dim NCLangFilePath As String
NCLangFilePath = GetTheRegValue(HKEY_LOCAL_MACHINE, "SYSTEM\CurrentControlSet\Services\PMAC\Device0\Nc0\System", "NCLanguageFile", "C:\NCLangEnglish.txt")
If (NCLanguageCtrl.LoadLanguageText(NCLangFilePath) = True) Then
  btnBack.Caption = NCLanguageCtrl.GetObjectCaption("frmDiagnosticMenu", "btnBack")
  btnPrmVar.Caption = NCLanguageCtrl.GetObjectCaption("frmDiagnosticMenu", "btnPrmVar")
  btnTerminal.Caption = NCLanguageCtrl.GetObjectCaption("frmDiagnosticMenu", "btnTerminal")
  btnPlot.Caption = NCLanguageCtrl.GetObjectCaption("frmDiagnosticMenu", "btnPlot")
  btnDataPages.Caption = NCLanguageCtrl.GetObjectCaption("frmDiagnosticMenu", "btnDataPages")
End If
```
In the above example, our NCLanguageX control is called NCLanguageCtrl. In case the NC language control is not able to find the language text, it will display all text in English. Users can modify the content of the NC language text file to reflect their own desired text. Please be careful to only modify the Caption Of the object section of the language text. If you modify the form name or the object name, then you should also modify the form name and the object name in NC5.0 project (by using the NC5.0 Designer Program) to see the changes. For example, if the user wishes to modify part of the NC language text (frmDiagnosticMenu) to show the caption in Japanese, it would be:

```xml
<frmDiagnosticMenu>
[btnBack :<<戻る]
[brnTerminal :ターミナル]
```

Currently NC5.0 supports English and Japanese versions, and in the near-feature we will support Chinese, Spanish, Russian and French languages.

**MSGS(F7) menu item**

By clicking on the menu item MSGS or pressing F7, the Message sub-menu form (frmErrorsMenu) will be displayed. Also, the Error list Form (frmErrorList) will be shown. The error list displays all the warning, errors and messages generated by NC.

Here is a description of the functions in the Messages sub menu. The BACK button will go back to the main menu. CLEAR MSGS will clear all the messages. RESET will reset all the errors and warnings. The following functions are used.

```xml
Scripts.Alarms.ClearAlarms
Devices.PmacNC1.ClearErrors
```

**Errors message Window (frmErrorList)**

The Errors message display window lists all the errors and warnings defined in errors.Dat file. The errors are queried through a Dquery control and the result is displayed on the error display window. This window can display 4 types of messages.

1) If the message is from NC and it is level 0, it will be displayed in Lime
2) If the message is a Warning from NC and it is level 1, it will be displayed in Yellow
3) If the message is a Stop message from NC and it is level 2, it will be displayed in Fuchsia
4) If the message is an Error from NC and it is level 3, it will be displayed in Red
The function **GetAllErrors** is used to get all the errors.

Axis Stop(Z): Motor on positive end of travel limit
Axis Stop(X): Motor open loop
Axis Stop(X): Motor on negative end of travel limit

**OPER(F8) menu item**

By clicking on the menu item OPER or pressing F8, the Operator sub-menu form (frmOperatorMenu) will be displayed. This option is only available if the user has enabled the software operator panel via the NC setup program. In addition, the Operator Form (frmoperator) will be shown.
Operator Window (frmOperator)

The software operator panel simulates the functionality of the hardware operator panel.

The following are the functions used for the software panel:

- SetMode(2)
- SetMode(1)
- Devices.PmacNC1.SetMode(3)
- Devices.PmacNC1.SetJogSelect(3)
- Devices.PmacNC1.SetJogSelect(2)
- Devices.PmacNC1.SetJogSelect(4)
- Devices.PmacNC1.SetJogSelect(1)
Section Two

Devices.PmacNC1.Command(CNC_SELDISTX1,0,0)

Devices.PmacNC1.Command(CNC_SELDISTX10,0,0)

Devices.PmacNC1.Command(CNC_SELDISTX100,0,0)

Devices.PmacNC1.Command(CNC_SELDISTX1000,0,0)

Devices.PmacNC1.Command(CNC_SELDISTX10000,0,0)

Devices.PmacNC1.SetAxisSelect(AxisIndex)

Devices.PmacNC1.Command(CNC_JOGPLUS,0,0)

Devices.PmacNC1.Command(CNC_JOGMINUS,0,0)

Query : GetSpindleOverride(0)

Cmd : SetSpindleOverride()

SetSpindleSelect(NC_SPINDLE_CCW)
To help the users with the common NC functions, we have implemented ALT-F shortcut keys. The following lists the keys and their functionality.

ALT+F2 : Devices.PmacNC1.Command(CNC_SINGLEBLOCKTOGGLE,0,0)
ALT+F3 : Devices.PmacNC1.Command(CNC_BLOCKDELETETOGGLE,0,0)
ALT+F4 : Devices.PmacNC1.Command(CNC_OPTIONSTOPTOGGLE,0,0)
ALT+F5 : Devices.PmacNC1.Command(CNC_USER1TOGGLE,0,0)
ALT+F6 : Devices.PmacNC1.Command(CNC_USER2TOGGLE,0,0)
ALT+F7 : Devices.PmacNC1.Command(CNC_USER3TOGGLE,0,0)
ALT+F8 : Devices.PmacNC1.Command(CNC_USER4TOGGLE,0,0)
ALT+F9 : Devices.PmacNC1.Command(CNC_USER5TOGGLE,0,0)
ALT+F10 : Devices.PmacNC1.Command(CNC_USER6TOGGLE,0,0)
ALT+F11 : Devices.PmacNC1.Command(CNC_USER7TOGGLE,0,0)
ALT+F12 : Devices.PmacNC1.Command(CNC_USER8TOGGLE,0,0)
SECTION 3: SAMPLES

In this section we will create a sample form in PHMI and add it to the NC project. We will add this form as part of the Diagnostic forms group. Please review carefully the following steps for creating this form and adding it to the diagnostic menu structure.

1. Open up an NC project in the PMAC-NC Designer program.

2. In the Forms, add a new form (you can do this by clicking on the + sign on top of the Forms group.

3. Right-click on the newly-added form and select the Runtime Properties.

4. In the General tab, change the form name to SampleForm.
5. In the Show tab, uncheck the system menu and title bar and then click OK.

6. From the Forms section, right-click anywhere in the Forms container and select Edit Groups.

7. In the Edit groups window, double-click on DiagGroup.

8. Add the Sample form to this group by clicking on the add button. We are adding sample form to the diaggroup, because we want our sample form to be part of the diagnostic group and also part of the diagnostic menu structure. When finished click ok on both windows.
9. Double-click on the sample form. This will open up a VBA form designer window, where you may add components and any Visual Basic programming logic code.
10. Change the form’s dimension to the following parameters. These values are the same as all other forms in the dynamic (not fixed) section of the NC program.
   a. Height = 324
   b. Left = 393.75
   c. Top = 159.75
   d. Width = 374.25

11. In the Sample Form, add a DButton activex control. In the custom properties of this button, change the button behavior to Toggle, change the press commands to Devices.Pmac1.P100=1 and the released commands to Devices.Pmac1.P100=-1. Also, change the Un-Pressed caption text to P1 = -1 and pressed caption text to P1= 1.

12. Add a DtextDisplay activeX to the form. In the custom properties of this text display, add the query to be Devices.Pmac1.P100. This text display will show the current value of P100.

13. Add a DLamp activeX control to the form. In the custom properties of this DLamp control, add the Un-Lit caption text to be p1=-1 and the lit caption text to be p1=1, add the query to be Devices.Pmac1.P100, change the lamp color to Green and the litlampcolor to Red.

14. Go back to the PHMI and double-click on frmDiagnosticMenu. Click on the button which corresponds to the F7 menu option and go to its custom properties. In the pressed command section, add the following command: Forms.DiagGroup.SampleForm.Show. This command will show the SampleForm and hide the previous visible form. Also, change the Un-Pressed caption text to Sample Form.

15. Run the NC project and then click on DIAG option from the NC’s main menu. While in Diag Mode, click on the Sample Form menu option and this will bring up the form you just created. To test the commands in the form, click on the button on the form and you should see the result.

16. To create any other form within the NC project, follow the above procedure and you can customize your NC based on your needs.