NOTE

Progress is an ongoing commitment at Giddings & Lewis. We continually strive to offer the most advanced products in the industry; therefore, information in this document is subject to change without notice. The illustrations and specifications are not binding in detail. Giddings & Lewis shall not be liable for any technical or editorial omissions occurring in this document, nor for any consequential or incidental damages resulting from the use of this document.

DO NOT ATTEMPT to use any Giddings & Lewis product until the use of such product is completely understood. It is the responsibility of the user to make certain proper operation practices are understood. Giddings & Lewis products should be used only by qualified personnel and for the express purpose for which said products were designed.

Should information not covered in this document be required, contact the Customer Service Department, Giddings & Lewis, 666 South Military Road, P.O. Box 1658, Fond du Lac, WI 54936-1658. Giddings & Lewis can be reached by telephone at (414) 921-7100.

401-54708-00 → M1014.0525

Version 1296

# Table of Contents

## Chapter 1 - PiCPage Setup

1.1 Introduction ................................................................. 1-1  
1.2 Components of a PiC System with PiCPage  
   Hardware Components .................................................. 1-4  
   PC Software Components ............................................. 1-4  
   Manuals ........................................................................ 1-4  
1.3 Getting Started ............................................................. 1-5  
   Start Up ........................................................................ 1-6  
   Study the Example ......................................................... 1-10  
   Design Pages ................................................................ 1-14  
   Merge Pages with LDO .................................................... 1-14  
   Run your LDO ................................................................ 1-15  

## Chapter 2 - Using the PiCPage Software

2.1 MS-DOS File Naming Conventions for PiCPage  
   Filename Extensions ....................................................... 2-1  
2.2 Page Composition - Regions ........................................... 2-2  
2.3 Types of Regions ............................................................ 2-4  
2.4 Size, Location, and Shape of Regions  
2.5 PiCPage Menus and Fields .............................................. 2-8  
2.6 Level 1 Main menu bar  
   Workstation .................................................................. 2-9  
   Files ............................................................................. 2-13  
2.7 Level 2 Menu bar  
   Workstation ................................................................ 2-15  
   Files (expanded) ............................................................ 2-16  
   Update tcs ..................................................................... 2-18  
   Region ........................................................................... 2-19  
   Display .......................................................................... 2-21  
   Set state ......................................................................... 2-21  
2.8 Level 3 Menu Bar Display  
   Display .......................................................................... 2-23  
   Set state ......................................................................... 2-23  
2.9 The Static Text Region  
   Static text field options ................................................ 2-24  
2.10 The Touch Key Region  
   Touch key field options .................................................. 2-29  
2.11 The Touch Button Region  
   Touch button field options ............................................. 2-30  
2.12 The Page Select Region  
   Page select field options ................................................ 2-33  
2.13 The Data Display/Entry Region ..................................... 2-34
Chapter 2 - The Display/Entry Region

2.14 The Bar Chart Region ...................................................... 2-37
   Bar chart field options .............................................. 2-38
   Setpoint Example ...................................................... 2-41
   Compare to variable Example ......................................... 2-42
2.15 The Dynamic Text Region ................................................ 2-42
   Dynamic text field options ......................................... 2-43
2.16 Help Screen ............................................................... 2-43
2.17 Creating Messages with Special Extended Characters ............. 2-44
   Creating Double Sized Text ......................................... 2-46
   Example from A099P014 ................................................. 2-46

Chapter 3 - Interfacing the Application LDO to Pages

3.1 Introduction ........................................................................ 3-1
3.2 Interfacing to the PAGEUDFB or PAGE2170 (Page.LDO or PAGE1.LDO) .. 3-4
3.3 Page Select portion of PAGEUDFB or PAGE2170 ....................... 3-5
3.4 Touch Button portion of PAGEUDFB or PAGE2170 ................... 3-8
3.5 Data Entry and Display portion of PAGEUDFB or PAGE2170 .......... 3-13
3.6 Touch Key portion of PAGEUDFB or PAGE2170 ....................... 3-22
3.7 File Display portion of PAGEUDFB or PAGE2170 ..................... 3-24
3.8 Dynamic Messages portion of PAGEUDFB or PAGE2170 ............. 3-26
3.9 TCS Keyboard portion of PAGEUDFB or PAGE2170 .................. 3-31

Index
Chapter 1 - PiCPage Setup

1.1 Introduction

PiCPage allows interactive development and maintenance of screens (pages) for a touch control screen (TCS) operator interface device used with the PiC. It is run on a workstation [MS-DOS based personal computer (PC)].

PiCPage has two essential components. One component is an executable workstation file (PiCPage.exe) that you use to create the pages for an application. It is also used to transfer page data to the TCS and to create files that are used by the PiC. This is explained in Chapter 2.

The other component is a User Defined Function Block* called PAGEUDFB for the 1031/1051 TCS or PAGE2170 for the 2170 ErgoTouch stored in the PAGE.LIB and used in the PAGE.LDO and PAGE1.LDO respectively. The inputs and outputs to the function blocks are labelled in Figure 1-2 and Figure 1-3. They perform many services, including communication control between the TCS and the PiC and data input processing. Your machine control application LDO and the UDFB LDO are incorporated into one LDO tying everything together.

PiCPage allows you to simulate the execution of the LDO on the touch control screen or on the workstation screen as pages are being developed. Variable values, text messages, entire pages or portions of pages can be displayed and refreshed on the touch screen during the programming process.

Page design is format free. Pages are composed of regions. A region is an area of the page which has a specific function, such as displaying a message. Several region types are available and there are few limitations on the sizes or locations of regions.

An application can have up to 255 pages. (This is dependent upon the amount of memory available in the TCS.) Page data is stored in MS-DOS files which can be copied for use in more than one application. Up to 255 applications can be created.

Some page examples are shown in Figure 1-1 below. You can run the example LDO to view a variety of page designs.

Figure 1-1. Page examples

* UDFBs are available with PiCPro version 4.0 and later. All the logic required to connect pages designed with PiCPage to your machine control application is contained in the UDFB. That logic was developed in the PAGEUDFB.LDO or PAGE2170.LDO (referred to as the source LDO), converted to the UDFB in PiCPro, and stored in the PAGE.LIB.

The PiCPro Software Manual contains more information on user defined function blocks.
Figure 1-2. PAGEUDFB for 1031/1051 TCS

Figures 1-2 and 1-3 illustrate the inputs and outputs on the UDFBs which control various parts of your PicPage application as indicated by the brackets. It ties your PicPage application into your machine control application. Chapter 3 provides more information on the UDFBs.

Figure 1-2 on this page is the PAGEUDFB used with models 1031 or 1051 TCS. Figure 1-3 on the next page is the PAGE2170 used with the 2170 ErgoTouch TCS. The only difference between the two is that in the PAGE2170 there is one additional input and one additional output. Otherwise they function identically.

The input is TR which allows you to use a touch button release. This feature is only available with the 2170 TCS. The output is the TSO which allows you to monitor communications to the TCS. This is important to do if you are using the TR. See Chapter 3 for an example,
1.2 Components of a PiC System with PiCPage

PiCPage is designed to be used with a PiC system.

Hardware Components

To use PiCPage, the following hardware is required:

- **PiC**: A PiC with RAMDISK on the CPU.
- **Touch Control Screen**: Monochrome or color TCS with memory
- **Workstation**: IBM XT or AT compatible with 640K bytes of memory and a serial port (same as required for PiCPro).
- **RS232C CABLES**:
  - 9-pin female to 9-pin female (supplied with PiCPro).
  - 9-pin male to 9-pin female (25-pin female for IBM XT).
  - 9-pin male to 10-pin terminal screw connector.
  - 25-pin female to 9-pin female.

PC Software Components

- **PiCPage**: For developing TCS pages to interface with application.
- **PiCServoPro**: For developing the machine control application LDOs for the PiC.
- **LDO Merge (optional)**: For combining LDOs.

Manuals

The following manuals will be helpful in working with PiCPage:

- PiCPage Manual
- PiC900 Hardware, PiCPro Software, and Function/Function Block Manuals
- TCS Manual
- LDO Merge Manual
1.3 Getting Started

An overview of the suggested steps to follow when working with PiCPage is listed below. Further explanation of these steps follows.

| Start up | 1. Set up the TCS.  
|          | 2. Load the PiCPage software on your workstation. 
|          | 3. Connect the TCS to the workstation. 
|          | 4. Run PiCPage on the workstation. |
| Study Example | 5. Open the EXAMPLE (A099) pages provided. 
|          | 6. Download all the pages to the TCS. 
|          | 7. Connect the TCS to User Port 2 on the PiC and connect the workstation to PiCPro Port 1 on the PiC. 
|          | 8. Run PiCPro on the workstation. 
|          | 9. Open and download the EXAMPLE.LDO or the EXAMPLE1.LDO to the PiC. 
|          | 10. Send the A099.BN1 file to the PiC RAMDISK. 
|          | 11. Explore the example pages to see some of the possibilities for page design. 
|          | 12. Exit PiCPro. |
| Design Pages | 13. Disconnect the TCS from the PiC and connect it to the workstation. 
|          | 14. Run PiCPage on the workstation. |
|          | 15. Design the pages you want for your application. 
|          | 16. Copy all the pages to the TCS. 
|          | 17. Build a binary (BN) file to transfer pages to the RAMDISK on the PiC. 
|          | 18. Exit PiCPage. |
| Merge Pages with LDO | 19. Run PiCPro. 
|          | 20. Use the PAGEUDFB or the PAGE2170 function block to integrate your pages with your application ladder logic. This can be approached in one of three ways: 
|          | 1. Create your application LDO starting with the PAGE.LDO or PAGE1.LDO. 
|          | 2. If you already have an application LDO, merge the PAGE.LDO or PAGE1.LDO with it using the LDO Merge software. 
|          | 3. Enter the PAGEUDFB or PAGE2170 into your application ladder. NOTE: This requires that you declare all the inputs and outputs to the function block in your software declaration table and is not recommended. PAGE.LDO and PAGE1.LDO have all the inputs and outputs declared for you. |
| Run Your LDO | 21. Reconnect the TCS to User Port 2 on the PiC. 
|          | 22. Turn the PiC key to stop and cycle power. 
|          | 23. Transfer your page binary file to the PiC RAMDISK using PiCPro. 
|          | 24. Download your application LDO to the PiC. |
NOTE

Throughout these steps, cable connections must be changed depending on what you are doing.

When using PiCPage, connect the TCS to the workstation.
When using PiCPro, connect the workstation to PiCPro Port 1 on the PiC CPU.
When running an application, connect the workstation to PiCPro Port 1 on the PiC CPU and connect the TCS to User Port 2 on the PiC CPU.

Start Up

1. Set up the TCS

Turn the TCS on and bring up the list of setup parameters on the TCS by pressing the setup button found in the back of the screen on the 1031/1051 or press anywhere outside the display area to bring up a menu from which to select setup on the 2170 ErgoTouch. Refer to the Touch Control Screen Manual for additional information on how to set and save parameter values.

Setup parameters should be as follows:

<table>
<thead>
<tr>
<th>For the 1031/1051 TCS</th>
<th>For the 2170 ErgoTouch TCS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background</td>
<td>Background normal</td>
</tr>
<tr>
<td>Foreground Color</td>
<td>Foreground Color 30*</td>
</tr>
<tr>
<td>Background Color</td>
<td>Background Color 40*</td>
</tr>
<tr>
<td>Display Activity</td>
<td>Display Activity on</td>
</tr>
<tr>
<td>Cursor Type</td>
<td>Cursor Type (blinking)</td>
</tr>
<tr>
<td>NRC</td>
<td>NRC off</td>
</tr>
<tr>
<td>Address</td>
<td>Address RS232</td>
</tr>
<tr>
<td>Baud Rate</td>
<td>Baud Rate 9600</td>
</tr>
<tr>
<td>Data Bits</td>
<td>Data Bits 7</td>
</tr>
<tr>
<td>Parity Enable</td>
<td>Parity Enable on</td>
</tr>
<tr>
<td>Parity Sense</td>
<td>Parity Sense even</td>
</tr>
<tr>
<td>Stop Bits</td>
<td>Stop Bits 2</td>
</tr>
<tr>
<td>XON/XOFF</td>
<td>XON/XOFF on</td>
</tr>
<tr>
<td>Local Echo</td>
<td>Local Echo off</td>
</tr>
<tr>
<td>Local/Remote</td>
<td>Local/Remote remote</td>
</tr>
<tr>
<td>New Line</td>
<td>New Line off</td>
</tr>
<tr>
<td>Auto Wrap</td>
<td>Auto Wrap on</td>
</tr>
<tr>
<td>Refresh Rate</td>
<td>H/W Handshake none</td>
</tr>
<tr>
<td></td>
<td>Transmit mode normal</td>
</tr>
</tbody>
</table>

* - optional, user's preference
2. Load the PiCPAGE software on your workstation

The PiCPAGE software comes on two disks. Make a backup copy of these disks and store the originals in a safe place. You will need to create a PiCPAGE directory on your system to store some of the files as indicated in the charts below. Place the remaining files listed in the directory that contains your LDOs. Refer to the chart that applies to the type of TCS you are using.

**PiCPAGE Files for Use with 1031/1051**

<table>
<thead>
<tr>
<th>Copy to PiCPAGE directory</th>
<th>Copy to LDO directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>PICPAGE.EXE</td>
<td>PAGE.LDO</td>
</tr>
<tr>
<td>PAGEMSGS.DAT</td>
<td>PAGE.REM</td>
</tr>
<tr>
<td>A099.AP1 files</td>
<td>PAGEUDFB.LDO*</td>
</tr>
<tr>
<td>A099.BN1</td>
<td>PAGEUDFB.REM</td>
</tr>
<tr>
<td></td>
<td>PAGE.LIB*</td>
</tr>
<tr>
<td></td>
<td>EXAMPLE.LDO</td>
</tr>
<tr>
<td></td>
<td>EXAMPLE.REM</td>
</tr>
<tr>
<td></td>
<td>FILE1.TXT</td>
</tr>
<tr>
<td></td>
<td>FILE2.TXT</td>
</tr>
</tbody>
</table>

**PiCPAGE Files for Use with 2170**

<table>
<thead>
<tr>
<th>Copy to PiCPAGE directory</th>
<th>Copy to LDO directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>PICPAGE.EXE</td>
<td>PAGE1.LDO</td>
</tr>
<tr>
<td>PAGEMSGS.DAT</td>
<td>PAGE1.REM</td>
</tr>
<tr>
<td>A099.AP1 files</td>
<td>PAGE2170.LDO*</td>
</tr>
<tr>
<td>A099.BN1</td>
<td>PAGE2170.REM</td>
</tr>
<tr>
<td></td>
<td>PAGE.LIB*</td>
</tr>
<tr>
<td></td>
<td>EXAMPLE1.LDO</td>
</tr>
<tr>
<td></td>
<td>EXAMPLE1.REM</td>
</tr>
<tr>
<td></td>
<td>FILE1.TXT</td>
</tr>
<tr>
<td></td>
<td>FILE2.TXT</td>
</tr>
</tbody>
</table>

**IMPORTANT**

The PAGE.LIB containing the PAGEUDFB and PAGE2170 function blocks and the source ladder PAGEUDFB.LDO or PAGE2170.LDO must always be in the same directory.

*It is essential that these two files be kept in the same directory.*
### PiCPage Directory

<table>
<thead>
<tr>
<th>File</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PICPAGE.EXE</td>
<td>Workstation development tool used to develop display pages.</td>
</tr>
<tr>
<td>PAGEMSGS.DAT</td>
<td>File containing all the help messages.</td>
</tr>
<tr>
<td>NOTE: Workstation computer setup information is stored in PiCPage.CFG. When PiCPage is run, the PiCPage.CFG file is created (if it does not already exist).</td>
<td></td>
</tr>
<tr>
<td>A099.AP1 files</td>
<td>Example pages demonstrating a variety of page designs available with PiCPage.</td>
</tr>
<tr>
<td>A099.BN1</td>
<td>The binary file for the EXAMPLE pages that must be sent to the PiC RAMDISK.</td>
</tr>
</tbody>
</table>

### LDO Directory

<table>
<thead>
<tr>
<th>File</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE.LDO or PAGE1.LDO</td>
<td>This ladder contains the PAGEUDBF or the PAGE2170 with all the inputs and outputs connected and declared in the software declaration table. Add your application LDO to this one. The application LDO networks can be placed before or after the executive networks.</td>
</tr>
<tr>
<td>PAGE. REM or PAGE1.REM</td>
<td>The REM file contains documentation for the LDO file. PiCPro will merge the REM file with the LDO file.</td>
</tr>
<tr>
<td>PAGEUDBF.LDO or PAGE2170.LDO</td>
<td>The PAGEUDBF.LDO or the PAGE2170.LDO is the source ladder for the UDFBs. Do not alter these LDOs in any way. When you are running any LDO containing the PAGEUDBF or the PAGE2170 you can view this source ladder from the View menu of PiCPro.</td>
</tr>
<tr>
<td>PAGEUDBF. REM or PAGE2170.REM</td>
<td>The REM file contains documentation for the LDO file. PiCPro will merge the REM file with the LDO file.</td>
</tr>
<tr>
<td>PAGE.LIB</td>
<td>The library that contains the PAGEUDBF and the PAGE2170 function blocks. Allows you to access the function blocks from the Function menu in PiCPro when building a network.</td>
</tr>
<tr>
<td>EXAMPLE.LDO or EXAMPLE1.LDO</td>
<td>Example pages. Use these for a reference or adapt them into the application LDO.</td>
</tr>
<tr>
<td>EXAMPLE. REM or EXAMPLE1.REM</td>
<td>The REM file contains documentation for the LDO file. PiCPro will merge the REM file with the LDO file.</td>
</tr>
<tr>
<td>FILE1.TXT FILE2.TXT</td>
<td>ASCII files for the EXAMPLE.LDO or the EXAMPLE1.LDO that must be sent to the PiC RAMDISK.</td>
</tr>
</tbody>
</table>
3. Connect the TCS to the workstation

Using the two cables shown below, make the connections in Figure 1-4 in order to run PiCPage on the workstation.

Figure 1-4. TCS to Workstation

Pin Out for Cables 1 and 2

<table>
<thead>
<tr>
<th>9-pin male (to PC serial port)</th>
<th>9-pin female (to PC serial port)</th>
<th>(25-pin female)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 to RD</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3 to TD</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>5 to GND</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6 to DSR</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>8 to CTS</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>9-pin female (to TCS serial port)</th>
<th>25-pin female (to TCS serial port)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 to RD</td>
<td>2</td>
</tr>
<tr>
<td>3 to TD</td>
<td>3</td>
</tr>
<tr>
<td>5 to GND</td>
<td>7</td>
</tr>
<tr>
<td>6 to DSR</td>
<td>11</td>
</tr>
<tr>
<td>8 to CTS</td>
<td>20</td>
</tr>
</tbody>
</table>
4. Run PICPage on your workstation

From the PICPAGE directory, type PICPAGE at the prompt and press <Enter>. This brings up the introductory copyright screen as shown below. It is displayed for a few seconds and then clears (or press any key to clear it).

---

**PICPage**

**PIC900 PicPage**

**Version 2.1**

Copyright (c) 1989-1992 Giddings & Lewis, Inc.

All Rights Reserved.

---

Study the Example

5. Open the EXAMPLE (A099) pages provided

Press F to bring up the options available under the Files menu as shown below.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New</td>
</tr>
<tr>
<td></td>
<td>Open</td>
</tr>
</tbody>
</table>

Press O to open the existing EXAMPLE file.

The Open file box appears with the default "1" in the Application and Page number boxes.

Enter the application number 99 and leave the page number at the default 1, press <F10>.

---

Open File

Application number: 99

Page number: 1

---

When you open an existing file like EXAMPLE the screen displays all the region numbers and types created for that page.
6. **Download all the pages to the TCS**

Press U to bring down the Update TCS menu.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>Display</td>
<td>Set state</td>
</tr>
<tr>
<td>............</td>
<td>To add a</td>
<td>Copy to tcs page memory</td>
</tr>
<tr>
<td>&lt;End of region table&gt;</td>
<td></td>
<td>copy All to page memory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Show tcs memory status</td>
</tr>
</tbody>
</table>

Press A to copy All to tcs page memory for the EXAMPLE application. (This replaces any existing pages in the TCS memory.)

NOTE: If the TCS is not connected to the workstation, you will get the following message when you attempt to send information to the TCS.

```
Error! No Touch Screen Connected.
Press any key to continue.
```

7. **Connect the TCS and the workstation to the PiC**

Now make all the cable connections as shown in Figure 1-5 connecting the PiC to both the workstation and the TCS.
Figure 1-5. TCS and workstation to PIC

Pin Out for PICPro Cable and Cable 3

<table>
<thead>
<tr>
<th>PICPro Cable</th>
<th>Cable 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>9-pin female</td>
<td>9-pin female connector</td>
</tr>
<tr>
<td>to Port 1:</td>
<td>to Port 2:</td>
</tr>
<tr>
<td>3 to RD</td>
<td>2 to RD</td>
</tr>
<tr>
<td>2 to TD</td>
<td>3 to TD</td>
</tr>
<tr>
<td>5 to GND</td>
<td>5 to GND</td>
</tr>
<tr>
<td></td>
<td>10-pin</td>
</tr>
<tr>
<td></td>
<td>connector</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>
8. Run PiCPro on the workstation

From your LDO directory, type PICPRO at the prompt.

9. Open the EXAMPLE.LDO or the EXAMPLE1.LDO to the PIC

Press M(odule), O pena, and <F8> to bring up the list of available LDOs. Select EXAMPLE or EXAMPLE1 by highlighting it and pressing <Enter>. Press <F10>
The EXAMPLE.LDO or EXAMPLE1.LDO is now on your screen.

10. Send the A099.BN1 file to the PiC RAMDISK

Load the EXAMPLE pages to the PiC RAMDISK by pressing P(rocessor), D(isk operations), and S(end file). Type in the path (wherever the PiCPage directory is located) and the file name (A099.BN1) in the Workstation File box.

Move the cursor to the To RAMDISK File: box. Type PAGE.BN1 and press <Enter>.

The Send File to Control RAMDISK box should look like the one shown below.

```
Send File to Control RAMDISK:

<F8-Select List>
Workstation File A099.BN1
To RAMDISK File PAGE.BN1

Cancel <Esc> Accept <F10>
```

Press <F10>.

If you get an error message, turn the PiC and the scan switch off. Restart the PiC and go through step 10 again.

11. Download the EXAMPLE.LDO or the EXAMPLE1.LDO to the PiC

Download the LDO to the PiC.
Press M(odule), D(ownload).

Start the scan on the PiC by turning the scan switch on. If the scan switch is in the Run position, the scan can also be started in PiCPro by typing P(rocessor), S(can control), and C(old restart).

12. Explore the EXAMPLE pages

The Main Menu showing 16 different touch buttons appears on the TCS. Spend some time going through the pages.

The comments on the networks in the EXAMPLE.LDO or the EXAMPLE1.LDO may be helpful in understanding how the pages are set up.

13. Exit PiCPro

When you have finished looking over the example pages, exit PiCPro. Press W(orkstation), R(eturn to DOS), and <Enter>.
Design Pages

Now you can begin to design your own pages. Chapter 2 explains how the PiCPage software works.

14. Disconnect the TCS from the PIC and connect it to the workstation

   Connect the TCS to the workstation as shown in Figure 1-4.

15. Run PICPage on the workstation

   From the PICPAGE directory, type PICPAGE at the prompt and press <Enter>.

16. Design the pages you want for your application

   Refer to the Chapter 2 for explanations of how to use all the features of PiCPage.

17. Copy all the pages to the TCS

   When you have finished designing all the pages for your application, copy them to the TCS from the Update menu using the command `copy All to tcs memory`.

18. Build a binary (BN) file to transfer pages to the RAMDISK

   Build the binary file from the Files menu with the `Build` command.

19. Exit PICPage

   Exit PiCPage with `W(orkstation), R(eturn to DOS), Y(es)`. 

Merge Pages with LDO

20. Run PICPro

   Type PiCPro to bring it up on the workstation.

21. Using the PAGE.LDO or the PAGE1.LDO

   Refer to Chapter 3 on how to combine the PAGE.LDO or the PAGE1.LDO with your machine control application.
Run your LDO

When you have finished combining your pages with your LDO, you are ready to run the application.

22. Reconnect TCS to User Port2 on the PIC

Refer to Figure 1-5.

23. Cycle power

Cycle power to the PiC with the key in the Stop position.

24. Send your binary file to the PiC RAMDISK

Send the binary file for your application to the PiC RAMDISK as PAGE.BIN.

25. Download your application LDO to the PiC

Start the scan by turning the PiC key to the run position.
Notes

To update existing page:BND file in RAMOISK. User must turn off scan & cycle power. This will allow old version BND to be deleted so new version can be copied to RAMOISK.

At start of scan, the BND is opened and kept open even of scan control is off. If scan off the file is still open. Must cycle pur (w/ scan on) to close file.
Chapter 2 - Using the PiCPage Software

You use the PiCPage software to design how the screens or "pages" will appear on the TCS. Each application can contain up to 255 pages. Page data is stored in DOS files. Files can be copied for use in more than one application.

Up to 255 applications can be developed. One application at a time is loaded into the PiC900 and the TCS.

2.1 MS-DOS File Naming Conventions for PiCPage

Workstation files describing the pages are created when you execute a ‘save’ or ‘save As’ in PiCPage. PiCPage uses a fixed filename convention. The filenames are organized by application number and page numbers.

Figure 2-1. File naming for PiCPage

PiCPage application files are automatically named as follows:

"AxxxPxxx.AP1"

where:
Axxx Application number from 001 to 255
Pxxx Page number from 001 to 255
.AP1 extension for a page data file

For example, the data describing page 5 of application 10 is stored in "A010P005.AP1".

NOTE: Since the EXAMPLE already is using application number 99, do not use this number in naming any of the applications you create.
Filename Extensions

PiCPage generates files with the following extensions:

.AP1 Page data.

.ABK Backup of previous .AP1 file - generated every time a File Save is performed and an .AP1 file already exists.

.LST A printable ASCII file which lists all of the information contained in an .AP1 file - generated with the PiCPage Print command.

.BN1* Contains all the information needed by the PiC900 at run time. This file must be built using the Build command in PiCPage and sent to PAGE.BN1 on the PiC900 RAMDISK.

.ASC* An ASCII file which may be used to "load" the page memory of the TCS using DOS without PiCPage - generated with the Generate command in PiCPage.

To copy an ASCII file to the TCS, enter two commands (from DOS):

Mode COM1: 9600,e,7,2,p
Copy Axxx.ASC COM1

* These files do not have a page number associated with them. For example, the filenames for application 5 would be "A005.BN1" and "A005.ASC".

2.2 Page Composition - Regions

Pages are composed of regions. (See Figure 2-2.) A region describes an area of the display screen and defines what it will be used for. A region is an area of the screen in which data is displayed or entered.

To create a page you create and map one or more regions. Each page can have up to 100 regions.

For every region on a page, you establish its size, location, and the type of data that is displayed or is to be entered by you. In addition you can assign various attributes to the regions.

Attributes refer to the means of displaying characters on the touch control screen. Attributes include: highlight, underline, blink, reverse, conceal, foreground color, background color, and conceal color.
Figure 2-2. Regions on a page (page 14 from the EXAMPLE)

On this page, sixteen touch buttons are bit-packed into variables which are displayed in decimal, hexadecimal, and binary formats.
2.3 Types of Regions

Each page can be comprised of from 1 to 100 regions. Various limits exist for the number of regions per page for any given type of region. The seven types of regions are summarized here.

1. **Static text display** - up to 100 per page
   Displays a text message. Attributes, text, and outlines can be set.
   *Static* refers to display information which is stored in the TCS page memory and will not be changed by the PiC900 control. The information is recalled from the TCS memory when a page is selected.

   As an example, a group of these regions can be used to form a list of options.

2. **Dynamic text display** - up to 100 per page
   Displays a text message. Attributes, text, outlines, and strings can be set.
   *Dynamic* refers to display information which is stored in the Page.BN1 file in the PiC900 RAMDISK. Data is displayed only when your application program requests its display.

   Dynamic text messages are unique in that they can be displayed on any page whereas all of the other region types can be displayed only when the page number they are programmed under is active. Up to six pages of dynamic text messages may be defined providing up to 600 dynamic messages.

   As an example, a dynamic text display message can be used to display a warning message.

3. **Touch key** - up to 40 per page
   Defines a numeric or other unique touch key for a keypad (used for entry of numeric or other data as touch inputs). Attributes, text, and outlines can be set.

   The PAGEUDFB includes a numeric entry service which automatically processes data input. When a touch key is pressed, a flag indicating the touch is energized for use by your application program.

   Touch keys differ from touch buttons in that they are displayed in only one state. Touch buttons may be displayed in up to four states.

   As an example, a group of these regions can form a keypad that enables you to enter coordinates.
4. **Touch button** - up to 40 per page
   Replaces oil-tight machine pushbuttons. Attributes, text, and outlines can be set.

   Your application program may dynamically select and display one of four touch button states by simply energizing a coil. The four display states can cause both the attributes and the text for the touch button to change.

   When a touch button is pressed, a flag indicating the touch is energized for use by your application program.

   As an example, a touch button region can be used as a cycle start button.

5. **Page select** - up to 20 per page
   Selects different pages. Attributes, text, and outlines can be set.

   The PAGEUDFB includes a service which calls up pages. You simply define which page is to be called up when you create the region.

   Note that a page can also be called up from your application LDO.

   As an example, a page select region can be used to select a new page.

6. **Data display/entry** - up to 40 per page
   Displays data as it is being entered. Attributes, text, and outlines can be set.

   It allows the format and location of data display to be defined and allows for the setting of limits for data entry.

   Data can be displayed upon page selection or it can be refreshed whenever the data value changes.

   As an example, a data display/entry region can be used to display or enter an axis position.

7. **Bar/chart** - up to 40 per page
   Displays a vertical or horizontal bar chart representation of a variable value. Attributes, text, and outlines can be set.

   The bar may change attributes in two ways as the variable changes - as a percentage of a setpoint variable or in comparison to the value of a separate variable.

   Whenever the data changes, the bar is refreshed automatically.

   As an example, a bar chart region can be used to display changing temperatures.
2.4 Size, Location, and Shape of Regions

The size of a region is defined by the number of cells in the region.

A cell is the smallest division on the TCS. Each cell can display one character - alpha, numeric, or special (extended). A TCS has 1920 cells - 80 across by 24 down as shown in Figure 2-3.

**Size** Only touch regions have size restrictions. All other regions can be any size. Touch regions include touch keys, touch buttons, and page select regions.

**Location** The locations of regions on a page is format free. Except for touch regions, all regions can be placed at any location on a screen. Guidelines for the location of a touch region relate to the size of the region.

PiCPage provides a method for establishing the size and location of a region that saves you the task of determining coordinates. PiCPage enables you to move the cursor on the TCS to locate regions and it calculates the coordinates as the cursor moves.

**Shape** Regions are rectangular in shape and can be outlined.
Figure 2-3. Master cells for the TCS
PiCPage is a menu driven program. It displays a menu bar at the top of every screen. Options are chosen from the menu. A menu option is selected by typing the highlighted "hot" key for the option, or by moving the cursor using the arrow keys to the option and pressing <Enter>.

For the creating and editing regions, PiCPage presents "fields" in data entry tables on the screen. Data is entered into them by typing, or in some cases by selecting options from pop-up menus.

PiCPage is called up by typing picpage at the MS-DOS prompt. An introductory copyright screen is displayed for a few seconds and then clears (or press any key to clear it).

There are three levels of menu bars in PiCPage as shown below.

**Level 1** appears after the copyright screen

```
Workstation  Files
```

**Level 2** appears after a file is open.

```
Workstation  Files  Update tcs  Region  Display  Set state
To add a region, select insert command from the Region menu
```

**Level 3** appears when a region is being created/edited.

```
Display  Set state
```

```
Page: 1  Region: 1  Type: Static text #1
```

Each level will be discussed in this chapter along with the options and fields connected to it.
2.6 Level 1 Main menu bar

When the copyright screen clears, the Main Menu of PiCPage is displayed. The screen has two menu options; Workstation and Files. Screens displayed at later stages are expansions of this screen.

Workstation   Files

Workstation

PiCPage software requires certain information about your workstation in order to communicate between it and the TCS.

Press W to bring up the options available under the workstation menu as shown below.

Workstation   Files
  Help on PiCPage
  Computer Setup
  Printer Setup
  Return to DOS

Help on PiCPage

Press H¹.

A help message explaining the PiCPage help system using the <F9> key appears. Press any key to exit.

NOTE: Help screens are available throughout the PiCPage program. Use the arrow keys to place the cursor on the item you want help with and press <F9>.

* For each menu option described in this chapter, pressing the hot key when the menu list is displayed accesses the option. It is also possible to access the option by moving the cursor to the item and pressing <Enter>.
Computer Setup

Press C.

The Computer setup box appears.

---

Computer setup

- Color Display: Yes
- Remove Snow: No
- TCS/PiC900 connected to Com[1 or 2]: 1

Changes will be implemented with the next start of this program

---

Cancel
Accept

There are defaults for each entry. If all the defaults are correct, press <F10> to accept them and return to the Workstation menu.

If you need to change any defaults, do the following.

Color Display: If the workstation is in color, press Y to change the default.

Remove Snow: If you have a CGA adapter board and if the screen flickers during updates, press Y. NOTE: Using this option will slow screen updates.

TCS/PiC900 connected to: Specify which computer port will be used to communicate with the TCS and the PiC900.

Press <F10> to accept any changes.

Always restart PICPage to implement the changes.

Pressing <Esc> at any time will return you to the Workstation menu and leave the default values unchanged.
Press P.

The Printer setup box appears. Enter the information for your printer.

**Page Length/Width:** Defines the size of the printed page for PiCPage .LST files. The settings do not affect a PiCPage .LST file because PiCPage prints one screen per page (80 columns by 25 characters followed by a form feed).

**Printer Open/Close:** Allows you to send special characters to the printer before and after the material to be printed is sent. This gives you the option of using various modes that may be available with your printer. Consult your printer manual.

**Replace:** Allows you to replace the extended ASCII characters used in PiCPage with the following if your printer does not print IBM extended ASCII characters. Example:

- A "+" sign for all the symbols marking corners, a "!" point for the vertical bar (fifth from the bottom) a ":-" sign for the horizontal bar (fourth from the bottom) a "->" sign for the triangle at the bottom of the list.

Press <F10> to accept your printer setup.
Return to DOS

Press R.

If no changes have been made, the Exit PiCPage? box appears.
Choose Yes to return to the DOS prompt. Choose No to return to the PiCPage menus.

If changes have been made, this box appears.

Choose Yes to save all your changes and return to the DOS prompt.
Choose No to return to the DOS prompt without saving any changes.
Files

Press F to bring up the options available under the Files menu as shown below.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New</td>
</tr>
<tr>
<td></td>
<td>Open</td>
</tr>
</tbody>
</table>

New

Press N to create a new file.

The New File box shown below appears with the default "1" in the Application and Page number lines.

Enter an application number (1 to 255), press <Enter>.

Enter a page number (1 to 255), press <Enter>.

Press <F10> or <Enter> to create a new file.

```
<table>
<thead>
<tr>
<th>New File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application number: 1</td>
</tr>
<tr>
<td>Page number: 1</td>
</tr>
<tr>
<td>Cancel &lt;Esc&gt;</td>
</tr>
</tbody>
</table>
```

NOTE: If you attempt to create a file that already exists, an error message appears and you must enter a new number.

The .AP1 filename can be found at the bottom of the screen.
Open

Press O to open an existing file.

The Open file box appears with the default "1" in the Application and Page number lines.

Enter an application number (1 to 255), press <Enter>.
Enter a page number (1 to 255), press <Enter>.
Press <F10> or <Enter> to open an existing file.

When you open an existing file the screen displays all the region numbers and types created for that page.
The .AP1 filename can be found at the bottom of the screen.
2.7 Level 2 Menu bar

After a new file has been created or an existing one opened, the menu bar changes as shown below (new file example). The menu bar includes: Workstation, Files, Update tcs, Region, Display, and Set state.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
<th>Update tcs</th>
<th>Region</th>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
</table>

To add a region, select insert command from the Region menu

Workstation

The Workstation menu remains the same as the one on level 1. See the descriptions of the four options in the Level 1 section.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Help on PiCPage</td>
<td></td>
</tr>
<tr>
<td>Computer Setup</td>
<td></td>
</tr>
<tr>
<td>Printer Setup</td>
<td></td>
</tr>
<tr>
<td>Return to DOS</td>
<td></td>
</tr>
</tbody>
</table>
Files (expanded)

The Files menu has some additional commands added to it as shown below.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
<th>Update tcs</th>
<th>Region</th>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>To</td>
<td>New</td>
<td>insert</td>
<td>command from the Region menu</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Open</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Save</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>save As</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Close</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Build</td>
<td>{.bn1}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Print</td>
<td>{.lst}</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Generate</td>
<td>{.asc}</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

New
Same as described for level 1. Allows you to create a new file while working in another one. You will always be asked if you want to save any changes to the current file.

Open
Same as described for level 1. Allows you to open an existing file while working in another one. You will always be asked if you want to save any changes to the current file.

Save
Press S.
Saves the file that is currently open and creates a backup.
This message appears to let you know your file has been updated.

File updated
Press any key to continue
**save As**
Press A.

Saves the file that is currently open under a different application and page number. The File Save As box appears.

NOTE: If a file already exists with the same number, you will be asked if you want to replace it with the one you are saving.

Enter the new number(s) (1 to 255) and press <F10>.

```
File Save As
Application number: 99
Page number: 1
[Cancel <Esc>] [Accept <F10>]
```

This is useful when creating duplicate pages. For example, it would allow you to duplicate a keypad for all data entry pages.

The "file updated" message appears to let you know the name has been changed and your file has been updated. Press any key to continue.

**Close**
Press C.

Closes and saves the file currently opened.

**NOTE**
The first five options, **New**, **Open**, **Save**, **save As**, and **Close**, affect the .AP1 file. When a save occurs, the previous .AP1 file is backed up as a .ABK file.

The remaining options, **Build**, **Print**, and **Generate**, create .BN1, .LST, and .ASC files respectively. NOTE: One of the pages of your application must be open in order to use these options.

**Build**
Press B.

Compiles the dynamic information from all of the pages which exist for this application and stores the data in the Axxx.BN1 file.

NOTE: If any changes have been made, they will be saved automatically.

If a dynamic message is greater than 255 characters, it will be truncated to 255 characters and the following warning message will appear.

```
Error! Truncating dynamic display message
Region = XX
Press any key to continue
```

XX is the region number.
Print

Press P.

Creates a AxxxPxxx.LST file of the currently open file which can then be printed.

Generate

Press G.

Creates a DOS ASCII file, Axxx.ASC, which can be used separately from PiCPage to load the pages into TCS memory.

To copy an ASCII file to the TCS, enter two commands (from DOS):

Mode COM1: 9600,e,7,2,p
Copy Axxx.ASC COM1

Update tcs

There are three options under the Update tcs menu as shown below.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
<th>Update tcs</th>
<th>Region</th>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>............</td>
<td>To add a</td>
<td>Copy to tcs page memory</td>
<td></td>
<td></td>
<td>from the 'Region' menu</td>
</tr>
<tr>
<td>&lt;End of region table&gt;</td>
<td>copy All to tcs page memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Show tcs memory status</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A TCS must be connected to your workstation in order to use these options. An error message will appear if no TCS is connected.

Copy to tcs page memory

Press C.

Resets the simulation settings to off, blanks and refreshes the touch control screen, and then copies the displayed page image to TCS memory.

copy All to tcs memory

Press A.

Copies all programmed pages for the application replacing existing pages in the TCS memory.

Show tcs memory status

Press S.

A box appears giving the status of the TCS memory. It will tell you how many pages and how much memory has been used and how much you have left depending on the memory in your TCS.

```
TCS Memory Status:
Pages used: 20
Pages left: 29
Memory used: 24832
Memory left: 36864
```

Press any key to continue.
Region

Press R to bring up the options available under the Region menu as shown below.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
<th>Update tcs</th>
<th>Region</th>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>............ To add a region, select</td>
<td>Edit &lt;enter&gt;</td>
<td>Insert &lt;ins&gt;</td>
<td>Copy</td>
<td>Delete &lt;del&gt;</td>
<td>from the 'Region' menu</td>
</tr>
<tr>
<td>&lt;End of region table&gt;</td>
<td>Find</td>
<td>Move</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE: The <Enter>, <Ins>, and <Del> keys can be pressed at any time while working in Level 2 to allow you to edit, insert, or delete a region without pulling down this menu. The hot keys listed with the options work when the menu is displayed.

Edit

Press E (or <Enter> with the cursor on the desired region) to edit an existing region.

The data entry tables for the specified region type appear. Edit any data and press <Enter> or <F10> to accept the changes. Press <F10> to return to level 2.

Insert

Press I (or <Ins> from the screen) to enter a new region. The Insert Region box appears with the number of the next available region entered.

Insert Region

Region Number: 16

Press F10 to accept the next available number or enter any unused region number. This latter option allows you to skip regions in order to organize the page.

Move the cursor to the region type you want to insert and press <Enter>. This brings up the level 2 menu bar with the data entry tables for the specified region type displayed. Creating/editing each region is done from this level by entering data in each field.

The fields for each region type are explained later in this chapter.
Copy
Press C to copy the region the cursor is on to the region you enter in the Copy to Region box.

Delete
Press D (or <Del>) with the cursor on the region to delete a region. The Delete Region box lists the region number and asks you to choose Yes or No.

Find
Press F to bring up the Find Region box. Enter the region you want to find and press <Enter>.

PiCPage scrolls to the specified region.

Move
Press M.
Deletes a region from its current location and moves it to the location specified.

Regions are not renumbered when a region is moved.
Display

Press D to bring up the options available under the Display menu as shown below.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
<th>Update tcs</th>
<th>Region</th>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>............</td>
<td>To add a region, select 'Insert' the 'Region' menu</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region</td>
<td>Page</td>
<td>Blank tcs</td>
<td>Simulate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Page</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blank tcs</td>
<td></td>
<td>Simulate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulate</td>
<td></td>
<td>Press B to clear the TCS.</td>
<td>Press S to simulate the entire page on the workstation screen. This allows you to preview your page design before a TCS is connected.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE: Pages are displayed in monochrome only. The bars in any bar charts are not displayed. Text designated as double high appears normal size but written twice and with a space between each character. Text designated as double wide appears with a space between each character. The extended character set does not appear.

Set state

Press S to bring up the options available under the Set state menu as shown below. When the cursor is on Set state, a message appears at the bottom of the screen to let you know if Dynamic text and Test data are activated.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
<th>Update tcs</th>
<th>Region</th>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>............</td>
<td>To add a region, select 'Insert' command f</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;End of region table&gt;</td>
<td>Touch/Bar state Dynamic text state Show test data state</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Using PICPage Software 2-21
**Touch/bar state**

Press T to bring up the Activate State box.

Enter a number from 1 to 4 in the box. The numbers correspond to the states (1 to 4) that can be defined for a touch button and/or bar region. (See the Touch button and Bar/chart region.)

![Activate State](image)

Press <F10> or <Enter> <Enter> to accept your choice.

**Dynamic text state**

Press D to bring up the Display Dynamic Text? box.

Choose Yes to turn dynamic text on or No to turn it off.

![Display Dynamic Text?](image)

**Show test data state**

Press S to bring up the Show Test Data? box. This allows test data entered in a field to be displayed on the TCS during simulation.

Choose Yes to turn this option on or No to turn it off.

![Show Test Data?](image)

**NOTE:** After turning the dynamic text on, display the text on the screen with the D(isplay), R(egion) command.
2.8  Level 3 Menu Bar Display

Display

Press D to bring up the options available under the Display menu on level 3 as shown below. These are the same commands available from the level 2 Display menu.

<table>
<thead>
<tr>
<th>Region</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>Blank tcs</td>
<td></td>
</tr>
<tr>
<td>Simulate</td>
<td></td>
</tr>
</tbody>
</table>

- **Region**: Press R to display the current region on the TCS.
- **Page**: Press P to display the entire page on the TCS.
- **Blank tcs**: Press B to clear the TCS.
- **Simulate**: Press S to simulate the entire page on the workstation screen. This allows you to preview your page design before a TCS is connected.

**NOTE:** Pages are displayed in monochrome only. The bars in any bar charts are not displayed. Text designated as double high appears normal size but written twice. Text designated as double wide appears with a space between each character. The extended character set does not appear.

Set state

Press S to bring up the options available under the Set state menu on level 3 as shown below. The first three commands are the same commands available from the level 2 Set state menu.

<table>
<thead>
<tr>
<th>Touch/bar state</th>
<th>Dynamic text state</th>
<th>Show text data state</th>
<th>Cursor mode state</th>
</tr>
</thead>
</table>

- **Touch/bar state**: Same as described for level 2.
- **Dynamic text state**: Same as described for level 2.
- **Show test data state**: Same as described for level 2.

When working in level 3 with creating/editing regions, a fourth command is added to the Set state menu whenever the cursor is on an applicable item in the data entry table. This is the Cursor mode state.
Cursor mode state  
Press C to activate the cursor mode.

A "Cursor Mode Active" message appears at the bottom of your PiCPage screen. The TCS cursor is now active and can be manipulated with the workstation arrow keys to define areas on the TCS.

Press S C to toggle the cursor mode off and return control of the cursor to the workstation.

2.9 The Static Text Region

The static text region is used to define those portions of a page which never change, or are static. The display image created by a static text region is stored in the TCS page memory and is not stored in the PAGE.BN1 file on PiC900 RAMDISK.

A page can have up to 100 regions. It can have up to 100 Static Text regions.

The Static Text field screen appears when you insert a region and select static text from the Insert box by pressing <Enter> with the cursor on static text.

<table>
<thead>
<tr>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page: 1</td>
<td>Region: 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locate region: 1</th>
<th>corner line: 1</th>
<th>corner column: 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box attributes: No</td>
<td>Outline region: No</td>
<td>size, line(s): 1</td>
</tr>
<tr>
<td>size, column(s): 1</td>
<td>style: 1</td>
<td>Text attributes: No</td>
</tr>
<tr>
<td>offset line(s): 0</td>
<td>offset column(s): 0</td>
<td>Text:</td>
</tr>
</tbody>
</table>

To create a Static Text Region, edit the fields in this screen and then save the changes by pressing <F10>.

To display on the TCS the changes you make to a field, choose D(isplay) R(egion).
Static text field options

Locate region
Defines the location of the upper left-hand cell of the region.

Corner line
Row number (1 to 24) of the upper left-hand cell of the region.

Corner column
Column number (1 to 80) of the upper left-hand cell of the region.

Numbers can be typed into these fields or PiCPPage can insert the numbers.
To have PiCPPage calculate and insert the corner line and column numbers:

1. Position the workstation cursor in either "corner" field.
2. Press S(et state) C(ursor mode state) to switch control of the cursor to the TCS.
3. Use the workstation arrow keys to move the TCS cursor. (The corner line field is updated if the cursor is moved vertically; the corner column field is updated if the cursor is moved horizontally.)
4. Press S C again to return control of the cursor to the workstation.

Box attributes
Assigns box attributes to the region. A submenu of options appears if the default value, NO, is changed to YES by pressing <Enter>, or if the value of the field is YES and the cursor is moved to this field with the down arrow key.

NOTE: The cursor always appears on the last item (conceal color) in this submenu. That is to let you exit the submenu with just one down arrow key rather than scrolling through all the options.

<table>
<thead>
<tr>
<th>highlight:</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>underline:</td>
<td>No</td>
</tr>
<tr>
<td>blink:</td>
<td>No</td>
</tr>
<tr>
<td>reverse:</td>
<td>No</td>
</tr>
<tr>
<td>conceal:</td>
<td>No</td>
</tr>
<tr>
<td>foreground color: white</td>
<td></td>
</tr>
<tr>
<td>background color: black</td>
<td></td>
</tr>
<tr>
<td>conceal color: black</td>
<td></td>
</tr>
</tbody>
</table>

If attributes are not assigned, the region is displayed with the default attributes shown in the submenu box above.

To edit any field, position the cursor on the option and press <Enter>.

NOTE: The conceal and conceal color are used together to cover an area. For example, you may want to conceal the region used to enter a password.
The <Enter> key toggles a NO/YES entry. The <Enter> key brings up the attribute color list for any color entry. Type the first letter of the color name to choose that color.

<table>
<thead>
<tr>
<th>Attribute colors</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLACK</td>
</tr>
<tr>
<td>RED</td>
</tr>
<tr>
<td>GREEN</td>
</tr>
<tr>
<td>YELLOW</td>
</tr>
<tr>
<td>BLUE</td>
</tr>
<tr>
<td>MAGENTA</td>
</tr>
<tr>
<td>CYAN</td>
</tr>
<tr>
<td>WHITE</td>
</tr>
<tr>
<td>GRAY</td>
</tr>
<tr>
<td>ORANGE</td>
</tr>
<tr>
<td>PEWTER</td>
</tr>
<tr>
<td>FUCHSIA</td>
</tr>
<tr>
<td>DARK BLUE</td>
</tr>
<tr>
<td>VIOLET</td>
</tr>
<tr>
<td>STEEL</td>
</tr>
<tr>
<td>KHAKI</td>
</tr>
</tbody>
</table>

NOTE: Color attributes have no effect if you have a model 1030 monochrome TCS.

Outline region
Outlines the region in one of four line styles. This field is used in conjunction with the next three fields. The value of this field must be YES or the outline will not display.

size, line(s)
Length, in cells, of the area of the region to be outlined. Numbers can be typed into these fields or PiCPage can insert the numbers by setting the cursor mode.

size, column(s)
Width, in cells, of the area of the region to be outlined. Numbers can be typed into these fields or PiCPage can insert the numbers by setting the cursor mode.

style
Defines the type of outline for the region. Four styles are available:

1. Single Line
2. Beveled Corners
3. Inside Margin
4. Double Line

Enter a number from 1 to 4.
**Text attributes**

Assigns attributes to the text in the region. Refer to Box attributes above. If YES is entered in this field but no attributes are assigned (from the submenus), the text will not display. If NO is entered in this field, the text will display with the attributes assigned for Box attributes.

**offset line(s)**

Number of cells down (from the upper left-hand corner of the region) to locate the first character of text.

**offset column(s)**

Number of cells to the right (from the upper left-hand corner of the region) to locate the first character of text.

The offset line(s) and column(s) options are useful to center or align text within the region.

Numbers can be typed into these fields or PiCPage will insert the numbers when you set the cursor mode.

**text**

Creates a text message for the region. Press <Enter>. A text window appears.

```
 Enter Message Text
```

The "position = 1" indicates the position of the cursor within the message. The "length = 0" indicates the total length of the message.

Text is typed in the window.

If you are editing an existing message:

Use the left/right arrow keys, and <Home>/<End> keys to position the cursor.

Use the <DEL> key to delete the character the cursor is beneath.

Use the <Backspace> key to delete the character to the left of the cursor.
Text messages may be up to 255 characters long. The text window displays 50 characters of the message. The message may be comprised of any alphanumerical or hexadecimal characters. Refer to section 2-17 for a description of creating messages with the extended character set.

Hexadecimal characters are entered as

$xx

where x = 0 to 9, or A to F.

For example, $1B is the hex representation of the escape character.

When text editing is terminated by pressing <F10>, all hex characters are evaluated and must be of the format "$x" followed by 2 hex characters. If an illegal hex character is detected, the workstation will beep and this message will be displayed:

Illegal Hex Character

NOTE: All characters (including $1B, for example) add to the length of the message.

A column of text messages can be created from one text message by pressing <Enter> at the point(s) where the next row should start. The text editor will automatically insert the necessary escape sequence for displaying a column of information. The sequence "$1B[aa;bbH" is inserted whenever <Enter> is pressed.

The system automatically replaces "aa" and "bb" with the appropriate touch screen row and column numbers when region editing is completed or when the region is displayed.

Press <F10> (never <Enter> since that inserts the escape sequence) to terminate text editing. Press <F10> again to save the message as edited.

Press ESC to restore the message to its previous state.
2.10 The Touch Key Region

The touch key region is used to create touch keypads that you can use to enter data.

A page can have up to 100 regions. It can have up to 40 touch key regions.

All touch regions have two sizes - the size of the region and the size of the touch area.

Touch areas must have sizes that are multiples of 8 * 2 (across by down). This is referred to as a master cell. There are 120 master cells on the TCS. Figure 2-3 at the beginning of this chapter shows the master cells for the TCS.

A touch key should have the same size and location as the touch area, but this is not required. However, if the touch area does not match the region area, there is the possibility that the touched area could be missed or that the area could be touched inadvertently.

The Touch Key Region screen appears when you insert a touch key region in level 2:

<table>
<thead>
<tr>
<th>Display Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page: 1 Region: 1 Type: Touch key #1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locate region:</th>
<th>Touch area:</th>
</tr>
</thead>
<tbody>
<tr>
<td>corner line: 1</td>
<td>map as key: '0'</td>
</tr>
<tr>
<td>corner column: 1</td>
<td>master cell: 0</td>
</tr>
<tr>
<td>Box attributes: No</td>
<td>height: 0</td>
</tr>
<tr>
<td>Outline region: No</td>
<td>width: 0</td>
</tr>
<tr>
<td>size, line(s): 1</td>
<td>auto-flash No</td>
</tr>
<tr>
<td>size, column(s): 1</td>
<td></td>
</tr>
<tr>
<td>style: 1</td>
<td></td>
</tr>
<tr>
<td>Text attributes: No</td>
<td>offset line(s): 0</td>
</tr>
<tr>
<td>offset column(s): 0</td>
<td>Text:</td>
</tr>
</tbody>
</table>

Using PICPage Software 2-29
**Touch key field options**

The first column of fields for this region are the same as those for the Static Text region. Refer to Static Text Region for information.

The second column fields are explained here.

**Touch area**

Defines the touch area of the region. The *touch area* is the area of the region that will "recognize" a touch.

**IMPORTANT:** A cell is the smallest division of the touch screen. Each cell can hold one character - either alpha, numeric, or special. A *master cell* is a group of 16 cells - 8 across by 2 down. Touch area boundaries must match master cell boundaries.

There are 120 master cells. Their locations are shown in Figure 2-3. The diagram is useful for mapping touch areas but it is not required. Touch areas can be mapped with the cursor.

**map as key**

Defines the key that is assigned to this region. When the cursor is in this field press <Enter> to see a list of keys:

<table>
<thead>
<tr>
<th>Touch Keys</th>
<th>Touch Keys</th>
<th>Touch Keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>+/-</td>
<td>tk[26]</td>
</tr>
<tr>
<td>1</td>
<td>ENT</td>
<td>tk[27]</td>
</tr>
<tr>
<td>2</td>
<td>RUB</td>
<td>tk[28]</td>
</tr>
<tr>
<td>3</td>
<td>DEL</td>
<td>tk[29]</td>
</tr>
<tr>
<td>4</td>
<td>NEXT</td>
<td>tk[30]</td>
</tr>
<tr>
<td>5</td>
<td>LAST</td>
<td>tk[31]</td>
</tr>
<tr>
<td>6</td>
<td>RIGHT</td>
<td>tk[32]</td>
</tr>
<tr>
<td>7</td>
<td>LEFT</td>
<td>tk[33]</td>
</tr>
<tr>
<td>8</td>
<td>tk[21]</td>
<td>tk[34]</td>
</tr>
<tr>
<td>9</td>
<td>tk[22]</td>
<td>tk[35]</td>
</tr>
<tr>
<td>.</td>
<td>tk[23]</td>
<td>tk[36]</td>
</tr>
<tr>
<td>+</td>
<td>tk[24]</td>
<td>tk[37]</td>
</tr>
<tr>
<td>-</td>
<td>tk[25]</td>
<td>tk[38]</td>
</tr>
</tbody>
</table>

The functions performed by TK 14 through 20 are:

- **ENT** enters data
- **RUB** backspace
- **DEL** zeros the displayed number
- **NEXT** chooses the next item
- **LAST** chooses the last item
- **RIGHT** chooses the item on the right
- **LEFT** chooses the item on the left

To assign a key to the region:

Use the arrow keys to move the cursor to a selection and press <Enter>.

Touch keys 0 to 20 are reserved for use by PAGEUDFB and by the Data Display/Entry region (to enable data entry).

Touch keys 21 to 39 can be defined by you.

Text must be entered for all keys, even if they are predefined. For example, if you are creating a delete key with TK(16), type DEL in the Text field in the first column. Or if you want TK(39) to be the letter Z, type Z in the Text field.
master cell
Defines the location of the upper left-hand cell of the touch area.

height
Height, in master cells, of the touch area.

width
Width, in master cells, of the touch area. The values for these fields can be typed in or entered automatically by PICPage.

auto-flash
Defines whether the area will flash when touched. Press <Enter> to change the value from NO to YES or from YES to NO.

2.11 The Touch Button Region

The touch button region is used to create a touch button that energizes a coil for use in your application LDO when it is touched. Four display states are available for touch buttons. Each state can cause the attributes and text of the region to change. A state is displayed by setting a control relay in the application LDO.

A page can have up to 100 regions. It can have up to 40 touch button regions.

All touch regions have two sizes - the size of the region and the size of the touch area. Touch areas must have sizes that are multiples of 8 x 2 (across by down).

A touch region should have the same size and location as the touch area, but it is not required. However, if the touch area does not match the region area, there is the possibility that the touched area could be missed or that the area could be touched inadvertently.

The Touch Button Region screen appears when you insert a touch button region in level 2.

<table>
<thead>
<tr>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page: 1 Region: 3 Type: Touch button #1</td>
<td></td>
</tr>
</tbody>
</table>

| Locate region: | 1 |
| corner line: | 1 |
| corner column: | 1 |
| Outline region: | No |
| size, line(s): | 1 |
| size, column(s): | 1 |
| style: | 1 |
| Touch area: | 0 |
| master cell: | 0 |
| height in cells: | 0 |
| width in cells: | 0 |
| auto-flash: | No |

| State 1 display: | No |
| Box attributes: | No |
| offset line(s): | 0 |
| offset column(s): | 0 |
| Text1: |
| State 2 display: | No |
| Box attributes: | No |
| offset line(s): | 0 |
| offset column(s): | 0 |
| Text2: |
| State 3 display: | No |
| Box attributes: | No |
| offset line(s): | 0 |
| offset column(s): | 0 |
| Text3: |
| State 4 display: | No |
| Box attributes: | No |
| offset line(s): | 0 |
| offset column(s): | 0 |
| Text4: |

A state can be displayed on the TCS as you are editing the region. This allows you to view the various states you are defining.
Press \textit{S}et state, \textit{T}ouch/bar state. Enter the number of the state you want to view on the TCS in the Activate State box and press \(<\text{F10}\>\).

Press \textit{D}isplay, \textit{R}egion to view the state you entered.

Repeat these commands to view another state.

**Touch/bar state**

Press \textit{T} to bring up the Activate State box.

Enter a number from 1 to 4 in the box. The numbers correspond to the states (1 to 4) that can be defined for a touch button and/or bar region. (See the Touch button and Bar/chart region.)

![Activate State](image)

Press \(<\text{F10}\>\) or \(<\text{Enter}\>\ <\text{Enter}\>\) to accept your choice.

\textbf{NOTE}: After setting a new state, display the region using the \textit{D}isplay, \textit{R}egion command.

**Touch button field options**

For the fields in column one and for the fields in column two except where noted below, see the corresponding fields of the Static Text and Touch Key regions.

\textbf{State 1 display}

Defines whether State 1 will display. State 1 is the default state when a page is activated. The value of this field must be \textit{YES} or \textit{no} button is displayed when a page is activated.

Press \(<\text{Enter}\>\) to change the value from \textit{NO} to \textit{YES} or from \textit{YES} to \textit{NO}.

The information defining state 1 is both static and dynamic. Static information is stored in the TCS page memory. Dynamic information is stored in the PAGE.BN1 file in the PiC900 RAMDISK so the state can be displayed properly if state 1 is reactivated.

\textbf{NOTE}: If a state attribute change is going to occur when the button is touched, the auto flash option should not be used. If you are changing the attributes for each state used, then all states defined should set attributes.

\textbf{text1}

Identical to the text field defined under Static Text Region, with an additional requirement. The length of the message is limited as listed below if State 1 display and/or Text attributes are \textit{YES}.

<table>
<thead>
<tr>
<th>length</th>
<th>Text attributes</th>
<th>State display</th>
</tr>
</thead>
<tbody>
<tr>
<td>247</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>202</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>202</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>
State 2 display  Defines the attributes for state 2 and whether state 2 will display.
State 3 display  Defines the attributes for state 3 and whether state 3 will display.
State 4 display  Defines the attributes for state 4 and whether state 4 will display.

States 2, 3, and 4 are identical to state 1 except they are optional.

Information defining states 2, 3, and 4 is dynamic only.

2.12 The Page Select Region

The page select region is used to enable you to select a new page.

A page can have up to 100 regions. It can have up to 20 page select regions.

All page select touch regions have 2 sizes - the size of the region and the size of the touch area.

Touch areas must have sizes that are multiples of 8 * 2 (across by down).

A touch region should have the same size and location as the touch area, but it is not required. However, if the touch area does not match the region area, there is the possibility that the touched area could be missed or that the area could be touched inadvertently.

The Page Select Region screen appears when you insert a page select region on level 2.

<table>
<thead>
<tr>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page: 1 Region: 4 Type: Page select #1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locate region:</th>
<th>Touch area:</th>
</tr>
</thead>
<tbody>
<tr>
<td>corner line: 1</td>
<td>page to select: 0</td>
</tr>
<tr>
<td>corner column: 1</td>
<td>master cell: 0</td>
</tr>
</tbody>
</table>

Box attributes: No
Outline region: No
size, line(s): 1
size, column(s): 1
style: 1

Text attributes: No
offset line(s): 0
offset column(s): 0
Text:
Page select field options

The first column of menu options and fields for this region are the same as those for the Static Text region. The second column of menu options and fields, except for page to select, are identical to the same fields of the Touch Key region. Refer to those regions for information.

Page to select

Defines the page that will display when you touch this area.

Enter a number from 1 to 255 and press <Enter>.

A page can also be "selected" from the application LDO. See Chapter 3.

NOTE

A page select region will always unconditionally select a new page. If conditions must be met before a new page is selected, use the page select from the ladder feature (see Chapter 3). A touch button region can be tied to this page select from the ladder so that touching the touch button will only select a new page after the conditions have been met.

2.13 The Data Display/Entry Region

The data display/entry region is used to enable you to display and enter variable values.

A page can have up to 100 regions. It can have up to 40 data display/entry regions.

The Data display screen appears when you insert a data/entry region in level 2.

<table>
<thead>
<tr>
<th>Page: 1</th>
<th>Region: 5</th>
<th>Type: Data display #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locate region:</td>
<td>corner line:</td>
<td>1</td>
</tr>
<tr>
<td>Box attributes:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Outline region:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>size, line(s):</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>size, column(s):</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>style:</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Text attributes:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>offset line(s):</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>offset column(s):</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Data display:</td>
<td>data variable #:</td>
<td>0</td>
</tr>
<tr>
<td># of digits:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>display sign:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>digits to right:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>refresh:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>double height:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>offset line(s):</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>offset column(s):</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Data entry limits:</td>
<td>upper:</td>
<td>0</td>
</tr>
<tr>
<td>lower:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Entry via buffer:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>corner line:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>corner column:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Next/last select:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>next:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>last:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>right:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>left:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Touch button select:</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>touch button:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Test data:</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
**Data display/entry field options**

The first column of the options and fields for this region are the same as those for the Static Text region. Refer to the options under Static Text Region for information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data display</strong></td>
<td>Defines the variable to display and the format of the display.</td>
</tr>
<tr>
<td><strong>data variable #</strong></td>
<td>Defines the variable number to be displayed in this region. Variables are double integers and are stored in the array DATA(0.255) in the PiC900. Variable 1 is DATA(1). Enter a number from 1 to 255 and press &lt;Enter&gt;.</td>
</tr>
<tr>
<td><strong># of digits</strong></td>
<td>Defines the variable’s total number of digits.</td>
</tr>
<tr>
<td></td>
<td>Enter a number from 1 to 32 and press &lt;Enter&gt;.</td>
</tr>
<tr>
<td></td>
<td>If the number of digits required to represent the variable is greater than the number of digits entered here, &quot;at&quot; signs (@) will print instead of the variable.</td>
</tr>
<tr>
<td><strong>display sign</strong></td>
<td>Defines whether a sign (+ / -) will be displayed.</td>
</tr>
<tr>
<td></td>
<td>Press &lt;Enter&gt; to change NO to YES or to change YES to NO.</td>
</tr>
<tr>
<td></td>
<td>If the value of the variable is negative and the value of this field is NO, then &quot;at&quot; signs (@) will print instead of data.</td>
</tr>
<tr>
<td><strong>digits to right</strong></td>
<td>Defines the number of digits to the right of the decimal point.</td>
</tr>
<tr>
<td></td>
<td>Enter a number from 0 to 10 and press &lt;Enter&gt;.</td>
</tr>
<tr>
<td>(0 = no decimal point)</td>
<td></td>
</tr>
<tr>
<td><strong>refresh</strong></td>
<td>Establishes whether the variable value will be refreshed whenever it changes (about every 5th scan if the variable changes continuously). If data entry is allowed, the value is refreshed as it is entered, even if this field is NO.</td>
</tr>
<tr>
<td></td>
<td>Press &lt;Enter&gt; to change NO to YES or to change YES to NO.</td>
</tr>
</tbody>
</table>
**double height**
Defines whether the variable will be displayed in a text size that is two cells high.
Press <Enter> to change NO to YES or to change YES to NO.
If you enter YES in this field you must also define double height via the text field in the first column, or by overlaying a static text region on top of this one. In the text editor box, enter two lines of escape sequences:

```
$1B#3$1B[aa;bbH
$1B#4
```

"$1B#3" defines the top half of the text.
"$1B[aa;bbH" gets inserted when you press <Enter> to create a second line.
"$1B#4 defines the bottom half of the text.

**offset line(s)**
Number of cells down to locate the first character of the variable.

**offset column(s)**
Number of cells to the right to locate the first character of variable.

**data entry limits**
Limits the range of values that can be entered by you for the variable.

**upper**
Enter a value from -2147483648 to +2147483647 and press <Enter>.

**lower**
Enter a value from -2147483648 to +2147483647 and press <Enter>.

**Test data**
Defines the data to display when interactively simulating.
Enter a constant value and press <Enter>.

**NOTE:** The data entered in the second column of the screen is dynamic data. It is compiled into the PAGE.BN1 file during file build.

This table shows some examples of data display.

<table>
<thead>
<tr>
<th>data displayed</th>
<th>actual data</th>
<th>display sign</th>
<th># of digits</th>
<th>digits to right</th>
</tr>
</thead>
<tbody>
<tr>
<td>1234</td>
<td>1234</td>
<td>no</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>+1234</td>
<td>1234</td>
<td>yes</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>+1.234</td>
<td>1234</td>
<td>yes</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>0.234</td>
<td>234</td>
<td>no</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>@@@@@</td>
<td>-234</td>
<td>no</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>@@@@@</td>
<td>123456</td>
<td>no</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

Data display may differ from "actual" because data in the PiC900 is integer only (no fractions).
Entry via buffer

Establishes whether data will display in a buffer area (a different area of the screen) as it is being entered by you. This feature may be used in conjunction with the next/last select or touch button select features.

If the value in this field is YES, the data will display in the area specified by the next two fields (corner line & corner column) until you press <Enter>. Then the data will display at the location specified by the region.

If the value in this field is NO, the data will display at the location specified by the region.

Press <Enter> to change NO to YES or to change YES to NO.

corner line

Defines the row number of the buffer area.

Enter a value from 1 to 24 and press <Enter>.

corner column

Defines the column number of the buffer area.

Enter a value from 1 to 80 and press <Enter>.

Next/last select

Enables you to enter data in another data entry region. A different region is selected when you touch a Next, Last, Right, or Left touch key. The regions are defined by the next four fields.

If next/last select is enabled, create Next, Last, Right, and Left touch keys also.

Press <Enter> to change NO to YES or to change YES to NO.

next

Establishes the data entry region that will be selected when this data entry region is selected and you touch the Next key.

Enter a value from 1 to 40 and press <Enter>.

last

Establishes the data entry region that will be selected when this data entry region is selected and you touch the Last key.

Enter a value from 1 to 40 and press <Enter>.

right

Establishes the data entry region that will be selected when this data entry region is selected and you touch the Right key.

Enter a value from 1 to 40 and press <Enter>.

left

Establishes the data entry region that will be selected when this data entry region is selected and you touch the Left key.

Enter a value from 1 to 40 and press <Enter>.

Touch button select

Defines whether data entry is activated for this region by a touch button. If YES, then you must touch the touch button defined by the next field (touch button) before you are able to enter data.

Press <Enter> to change NO to YES or to change YES to NO.

touch button

Number of the touch button region that enables data entry for this region.

Enter a number from 1 to 40 and press <Enter>.

2.14 The Bar Chart Region

The bar chart region graphs, in a bar chart, the changing value of a variable. The bar chart can be horizontal or vertical. The variable value can be charted relative to another variable. The "other" variable can be a setpoint variable or it can be a compare to variable. Or, two compare to variables can be used.
If the other variable is setpoint, the charted variable is charted based on a percentage of the setpoint value. If the other variable is compare to, the charted variable is charted based on the inequality of the two variables.

When the variable is charted relative to the value of another variable, the bar attributes (e.g., color) chart can be changed. They are changed by evoking different "states" when the percentage or (in)equality is reached. The states are like touch button states. But touch button states are changed from the application LDO whereas bar/chart states are changed by the PAGEUDFB.

A page can have up to 100 regions. It can have up to 40 bar/chart regions.

---

**Display Set state**

<table>
<thead>
<tr>
<th>Page: 1</th>
<th>Region: 6</th>
<th>Type: Bar chart #1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locate region:</td>
<td>Locate bar:</td>
<td>On % of setpoint:</td>
</tr>
<tr>
<td>corner line:</td>
<td>offset line(s):</td>
<td>No</td>
</tr>
<tr>
<td>corner column:</td>
<td>offset col(s):</td>
<td>Compared to variable: No</td>
</tr>
<tr>
<td>Box attributes:</td>
<td>Define bar:</td>
<td>State 1 data:</td>
</tr>
<tr>
<td>No</td>
<td>vertical:</td>
<td>attributes: No</td>
</tr>
<tr>
<td>Outline region:</td>
<td>horizontal:</td>
<td>% of setpoint, in</td>
</tr>
<tr>
<td>No</td>
<td>size:</td>
<td>tens of %: 0</td>
</tr>
<tr>
<td>size, column(s):</td>
<td>thickness:</td>
<td>compare variable #: 0</td>
</tr>
<tr>
<td>1</td>
<td>full scale:</td>
<td></td>
</tr>
<tr>
<td>style:</td>
<td>zero scale:</td>
<td></td>
</tr>
<tr>
<td>Text attributes:</td>
<td>Chart variable #: 0</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>text:</td>
<td>Test data: 0</td>
</tr>
<tr>
<td>offset line(s): 0</td>
<td>Setpoint variable #: 0</td>
<td></td>
</tr>
<tr>
<td>offset column(s): 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Text:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Bar chart field options

The first column of menu options and fields for this region are the same as those for the Static Text region. Refer to *Static Text Region* for information.

**Locate bar** Establishes the location of the bar.

**offset line(s)** Number of rows down, from upper left-hand corner of region, to locate bar.

**offset column(s)** Number of columns to the right, from upper left-hand corner of region, to locate bar.

See these fields for a Static Text region for information.

**Define bar** Defines the characteristics of the bar.

**vertical** Defines a vertical bar.

Press <Enter> to change NO to YES or to change YES to NO.
horizontal

Defines a horizontal bar.
Press <Enter> to change NO to YES or to change YES to NO.
NOTE: Either vertical or horizontal must be selected.

size

Defines the size, in cells, of the bar.
If the bar is vertical, this size is the maximum height of the bar.
Enter a value from 1 to 24 and press <Enter>.
If the bar is horizontal, this size is the maximum length of the bar.
Enter a value from 1 to 80 and press <Enter>.

thickness

Defines the thickness, in cells, of the bar.
If the bar is vertical, this size is the width of the bar.
Enter a value from 1 to 80 and press <Enter>.
If the bar is horizontal, this size is the height of the bar.
Enter a value from 1 to 24 and press <Enter>.

full scale

Establishes when the charting of the variable is complete. When the
variable reaches the value entered here, the bar will be 100% charted.
Enter a number from -2147483648 to +2147483647 and press
<Enter>.

zero scale

Establishes when the charting of the variable should begin. When the
variable reaches the value entered here, the charting of the bar will
start.
Enter a number from -2147483648 to +2147483647 and press
<Enter>.

Chart variable #

Defines the variable to be charted. The variable is from the
DATA(0...255) array.
Enter a number from 1 to 255 and press <Enter>.
(Index entry 0 is not used and is not a valid entry.)

Test data

Defines the data to display when interactively simulating.
Enter a constant value and press <Enter>.

Setpoint variable #

Defines the number of the variable to be used as a setpoint. The
variable is from the DATA(0...255) array.
Enter a number from 1 to 255 and press <Enter>.
(Index entry 0 is not used and is not a valid entry.)
If a variable number is entered here:
Enter YES in the on % of setpoint field for state 1 and/or state 2.
Enter a percentage in the % of setpoint in tenths of % field for state 1
and/or state 2.

NOTE: The information entered in the second column of the
screen is dynamic information. It is compiled into the PAGE.BN1
file during file build.
On % of setpoint
Establishes if attributes are based on a percentage of the setpoint variable value. If this field is YES then the compared to variable field should be NO. The percentage is entered in the % of setpoint in tenths of % field for states 1 and 2.
Press <Enter> to change NO to YES or to change YES to NO.

Compared to variable
Establishes if attributes are based on the compared to variable. If this field is YES then the previous field should be NO. The variable number is entered in the compare variable # field for states 1 and 2.
Press <Enter> to change NO to YES or to change YES to NO.

State 1 data
Defines the attributes of the bar for state 1.
If a setpoint variable has been defined, the attributes are in effect when the value of the chart variable is less than or equal to the percentage entered for this state (below) of the setpoint variable.
If a compare variable is defined, the attributes are in effect when the value of the chart variable is less than or equal to the value of the variable entered for this state (below).

attributes
Defines the attributes of the bar. Refer to this field for a Static Text Region for information.

% of setpoint in tenths of %
Defines the percentage, in tenths of a percent, for the setpoint variable. If a number is entered here, the compared to variable field should be NO and the on % of setpoint field should be YES.
Enter a number from 0 to 9999 and press <Enter>. (0 to 999.9%)

compare variable #
Defines the variable that the chart variable is to be compared to. The variable is from the DATA(0...255) array. If a number is entered here, the compared to variable field should be YES and the on % of setpoint field should be NO.
Enter a number from 1 to 255 and press <Enter>.
(Index entry 0 is not used and is not a valid entry).

State 2 data
Defines the attributes of the bar for state 2.
If a setpoint variable has been defined, the attributes are in effect when the value of the chart variable is less than or equal to the percentage of the setpoint variable for state 2 and greater than the percentage for state 1.
If a compare variable is defined, the attributes are in effect when the value of the chart variable is less than or equal to the value of the compared to variable for state 2 and greater than the value of the variable for state 1. The value of the state 2 variable should be larger than the value of the state 1 variable.

attributes
Defines the attributes of the bar. Refer to this field for a Static Text Region for information.

% of setpoint, in tenths of %
Defines the percentage, in tenths of a percent, for the setpoint variable. This value should be greater than the percentage for state 1. If a number is entered here, the compared to variable field should be NO and the on % of setpoint field should be YES.
Enter a number from 0 to 9999 and press <Enter>. (0 to 999.9%)
**compare variable #**
Defines the variable that the chart variable is to be compared to. The variable is from the DATA(0...255) array. If a number is entered here, the *compared to variable* field should be YES and the *on % of setpoint* field should be NO.

Enter a number from 1 to 255 and press <Enter>.

(Index entry 0 is not used and is not a valid entry).

**State 3 attributes**
Defines the attributes of the bar for state 3.

If a setpoint variable has been defined, the attributes are in effect when the value of the chart variable is greater than the percentage of the state 2 setpoint.

If a compare variable is defined for state 2, the attributes are in effect when the value of the chart variable is greater than the value of the state 2 variable.

If a state 2 variable is not defined, the state 3 attributes do not go into effect.

**Setpoint Example**
As an example of charting a variable with respect to a setpoint variable, assume the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Setpoint Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertical:</td>
<td>YES</td>
<td>Data (15) = 500</td>
</tr>
<tr>
<td>horizontal:</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>full scale:</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>zero scale:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>chart variable #:</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>setpoint variable #:</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>on % of setpoint:</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>compared to variable:</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>state 1 - percentage, in tenths of percent:</td>
<td>950 (95%)</td>
<td>State 3 525 &lt; DATA(10)</td>
</tr>
<tr>
<td>state 2 - percentage, in tenths of percent:</td>
<td>1050 (105%)</td>
<td>State 2 475 &lt; DATA(10) ≤ 525</td>
</tr>
<tr>
<td></td>
<td></td>
<td>State 1 DATA(10) ≤ 475</td>
</tr>
</tbody>
</table>
Compare to variable Example

As an example of charting a variable with respect to another variable, assume the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Compare Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>vertical:</td>
<td>NO</td>
<td>Data (11) = 400</td>
</tr>
<tr>
<td>horizontal:</td>
<td>YES</td>
<td>Data (12) = 600</td>
</tr>
<tr>
<td>full scale:</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>zero scale:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>chart variable #:</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>setpoint variable #:</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>on % of setpoint:</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>compared to variable:</td>
<td>YES</td>
<td></td>
</tr>
<tr>
<td>state 1 - compare variable #:</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>state 2 - compare variable #:</td>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

2.15 The Dynamic Text Region

The dynamic text region is commonly used to display machine states and fault messages.

The dynamic text region allows your application program to display text, change attributes, outline regions, and/or display strings generated in the Pic900. Usually this is done by using the display text feature only. But with the ability to control attributes, outlines, and string displays from the ladder, PicPage provides flexibility for complex display activity.

Dynamic text regions can be displayed on any page based on the state of the application. Up to six pages, with 100 dynamic text regions each, can be initialized when the application program starts scanning. Typically all of the dynamic messages are stored under unique page numbers that are used for that purpose only.

The Dynamic Text Region screen appears when you insert a Dynamic text region on level 2:
Display Set state

Page: 1  Region: 7  Type: Dynamic text #1

| Locate region: | corner line: 1 |
| Box attributes: | Yes |
| Outline region: | highlight: No |
| size, line(s): | underline: No |
| size, column(s): | blink: No |
| style: | reverse: No |
| Text attributes: | conceal: No |
| offset line(s): | foreground color: white |
| offset column(s): | background color: black |
| Text: | conceal color: black |

Dynamic text field options

The options and fields for this region are the same as those for the Static Text region with the exception of an additional requirement for text as explained below. Refer to Static Text Region for information.

Identical to the text field defined under Static Text Region, with an additional requirement - the length of the message is limited to less than 255 characters.

If set attributes and/or outline region are YES, the message length is limited as defined in the following table. Messages which are too large will be flagged during the file build.

<table>
<thead>
<tr>
<th>length</th>
<th>set attributes 1</th>
<th>outline region</th>
<th>set attributes 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>247</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>202</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>227</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>211</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>146</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

2.16 Help Screen

Help screens are available for menu items in PiCPage as you work through the program. If you want to read the help message, highlight the selection and press <F9>.
Creating Messages with Special Extended Characters

Messages can be created with "special" characters by typing the hexadecimal representations of the characters.

All hexadecimal characters must be entered in the following format:

$0E  Enters the special character mode.
$mm  Enters the special character.

The "$" signifies that a hex code follows and "mm" is the two digit hex code for the character.  (See Figure 2-4 for codes.)

$0F  Returns to the normal character mode.

Example

To display the following special characters on the TCS:

£ Û

Enter the following in the text box:

$0E$C$3$AA$0F

In determining message lengths, "$mm" counts as 3 characters.  Thus, the two character message in the example above is actually 12 characters long.

The hex codes in Figure 2-4 are found on the lower right of each character.
Figure 2-4. Special or extended character set
Creating Double Sized Text

To create text that is twice as high or wide (2 cells) as normal text, enter the following hexadecimal characters before the text:

| Double high top half:         | $1B#3 |
| Double high bottom half:      | $1B#4 |
| Example:                      | $1B#3 text message $1B#4 text message |
| Result:                       | text message |

Double high text must be entered on 2 lines, as shown. Otherwise only the top or bottom half of the text is displayed.

| Double width:                 | $1B#6 |
| Example:                      | $1B#6 text message |
| Result:                       | text message |

It is a requirement of the touch screen software that if double high or wide text is entered, the whole line of text (80 columns) must be double sized. Otherwise the text does not display properly.

To make an outlined region double sized or double wide, the entire region (including the area for the box outline) must be defined as double size or double wide. This can be done by overlaying a static text region to set the double size or including the proper codes in the text for the region.

Example from A099P014

Using page 14 from the EXAMPLE application, we can review how PiCPage is used.

1. Open Application number 99, page number 14 in PiCPage.
The screen in Figure 2-5 appears on your workstation. There are 22 regions defined for this page using four different types of regions.

NOTE: The numbers in the Type # column represent the number of times each region type has been used. For example, page 14 uses 16 touch buttons, two static texts, three data displays, and one page select.

**Figure 2-5. A099P014 in PICPage**

<table>
<thead>
<tr>
<th>Region</th>
<th>Files</th>
<th>Update texts</th>
<th>Region</th>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region 1</td>
<td>Static text</td>
<td>#1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 2</td>
<td>Static text</td>
<td>#2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 3</td>
<td>Touch button</td>
<td>#1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 4</td>
<td>Touch button</td>
<td>#2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 5</td>
<td>Touch button</td>
<td>#3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 6</td>
<td>Touch button</td>
<td>#4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 7</td>
<td>Touch button</td>
<td>#5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 8</td>
<td>Touch button</td>
<td>#6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 9</td>
<td>Touch button</td>
<td>#7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 10</td>
<td>Touch button</td>
<td>#8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 11</td>
<td>Touch button</td>
<td>#9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 12</td>
<td>Touch button</td>
<td>#10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 13</td>
<td>Touch button</td>
<td>#11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 14</td>
<td>Touch button</td>
<td>#12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 15</td>
<td>Touch button</td>
<td>#13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 16</td>
<td>Touch button</td>
<td>#14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 17</td>
<td>Touch button</td>
<td>#15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 18</td>
<td>Touch button</td>
<td>#16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 19</td>
<td>Data display</td>
<td>#1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 20</td>
<td>Data display</td>
<td>#2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 21</td>
<td>Data display</td>
<td>#3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 22</td>
<td>Page select</td>
<td>#1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<End of region table>
Figure 2-6 shows how page 14 from the EXAMPLE application appears on the TCS. The four types of regions being used are indicated.

![Diagram of Page 14 on TCS]

On this page, sixteen touch buttons are bit-packed into variables which are displayed in decimal, hexadecimal, and binary formats.
2. Move the cursor to Region 18 as shown in Figure 2-5 and press <Enter>. This brings up the screen shown in Figure 2-7. The information at the top of the page (encircled) lets you know that you are working with page 14, region 18, touch button 16.

You can see from the information in the first column how touch button 16 was defined according to location, outlining, and touch area.

Figure 2-7. EXAMPLE page 14 on TCS

<table>
<thead>
<tr>
<th>Display Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page: 14 Region: 18 Type: Touch button #16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locate region:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>corner line:</td>
<td>7</td>
</tr>
<tr>
<td>corner column:</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Outline region:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>size, line(s):</td>
<td>4</td>
</tr>
<tr>
<td>size, column(s):</td>
<td>6</td>
</tr>
<tr>
<td>style:</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Touch area:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>master cell:</td>
<td>32</td>
</tr>
<tr>
<td>height in cells:</td>
<td>2</td>
</tr>
<tr>
<td>width in cells:</td>
<td>1</td>
</tr>
<tr>
<td>auto-flash:</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State 1 display:</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box attributes:</td>
<td>Yes</td>
</tr>
<tr>
<td>offset line(s):</td>
<td>1</td>
</tr>
<tr>
<td>offset column(s):</td>
<td>1</td>
</tr>
<tr>
<td>Text1:</td>
<td>'Bit 16$1B[09,'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State 2 display:</th>
<th>Yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box attributes:</td>
<td>Yes</td>
</tr>
<tr>
<td>offset line(s):</td>
<td>2</td>
</tr>
<tr>
<td>offset column(s):</td>
<td>1</td>
</tr>
<tr>
<td>Text2:</td>
<td>'On'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State 3 display:</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box attributes:</td>
<td>No</td>
</tr>
<tr>
<td>offset line(s):</td>
<td>0</td>
</tr>
<tr>
<td>offset column(s):</td>
<td>0</td>
</tr>
<tr>
<td>Text3:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State 4 display:</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Box attributes:</td>
<td>No</td>
</tr>
<tr>
<td>offset line(s):</td>
<td>0</td>
</tr>
<tr>
<td>offset column(s):</td>
<td>0</td>
</tr>
<tr>
<td>Text4:</td>
<td></td>
</tr>
</tbody>
</table>

A099F014.AP1

BG79-2292

Using PICPage Software 2-49
3. The touch button is active in two states as shown in column two. (States 3 and 4 in column three are not used.) State 1 defines how touch button 16 will look when page 14 is selected. State 2 defines how it will look when it is touched.

Move the cursor to the Text1: line in column two and press <Enter>.

This brings up the Enter Text Message box shown in Figure 2-8. The text on touch button 16 when in State 1 has been entered here.

\[ \text{Bit 161} \]
\[ \text{Off} \]

NOTE: The encircled code is entered automatically by the software when a carriage return is entered.

**Figure 2-8. EXAMPLE page 14 on TCS**

<table>
<thead>
<tr>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page: 14</td>
<td>Region: 18</td>
</tr>
</tbody>
</table>

| Locate region: | 7 | 9 |
| Corner line: | | |
| Corner column: | | |

| Outline region: | 2 |
| Size, line(s): | 10 |
| Size, column(s): | 2 |

| Touch area: | 32 |
| Master cell: | 2 |
| Height in cells: | 1 |
| Width in cells: | 1 |
| Auto-flash: | No |

| State 1 display: | Yes |
| Box attributes: | Yes |
| Offset line(s): | 1 |
| Offset column(s): | 1 |

| State 3 display: | No |
| Box attributes: | No |
| Offset line(s): | 0 |
| Offset column(s): | 0 |

**NOTE:** Bit 16 is circled at position 10 in the outline region.

| Text2: | "On" |
| Box attributes: | Yes |
| Offset line(s): | 0 |
| Offset column(s): | 0 |

| Text4: | " " |
| Box attributes: | No |
| Offset line(s): | 0 |
| Offset column(s): | 0 |

4. When touch button 16 is touched, it goes to State 2 and takes on the attributes defined there. Note that the text changes to “On” (Text 2). The On is made to appear on the second line of the touch button by entering an offset line of “2” (encircled) in the Box attributes of State 2.
Chapter 3 - Interfacing the Application LDO to Pages

3.1 Introduction

The PAGEUDFB and the PAGE2170 (shown in Figure 3-1 for use with the 1031/1051 TCS and Figure 3-2 for use with the 2170 ErgoTouch TCS) link your TCS application to your machine control application. The machine control LDO must be combined with the PAGEUDFB or the PAGE2170. This can be accomplished in any of the following ways.

- If you have not yet created your machine control LDO, then open the PAGE.LDO (containing the PAGEUDFB) or the PAGE1.LDO (containing the PAGE2170) and add your machine control networks to it.

  NOTE: Use the save As command in PiPro to assign a new name to this application LDO keeping the original PAGE.LDO and PAGE1.LDO for future applications.

- If you already have a machine control LDO, use the optional LDO Merge software to combine it with PAGE.LDO or PAGE1.LDO.

NOTE: It is also possible to enter the PAGEUDFB or the PAGE2170 into a network of an existing LDO from the Functions menu:

(USER→PAGE→PAGEUDFB or PAGE2170).

However, this is not recommended since it means you must declare all the variables in the software declarations table and make all the connections to the PAGEUDFB or PAGE2170 making sure that each variable has exactly the same data type as shown in Figure 3-1 or 3-2. When you use PAGE.LDO or PAGE1.LDO, these tasks are done for you.
Figure 3-2. PAGE2170

Page Select

Page Select

Touch Button

Touch Button

Data Entry and Display

Data Entry and Display

Dynamic Messages

Dynamic Messages

TCS Keyboard

TCS Keyboard

TCS Monitor

TCS Monitor

Interfacing the Application LDO to Pages
3.2 Interfacing to the PAGEUDFB or PAGE2170 (Page.LDO or PAGE1.LDO)

When you set coils in the LDO or touch buttons on the TCS, the PAGEUDFB or the PAGE2170 interprets those commands. The PAGEUDFB or the PAGE2170 must be called every scan as shown in the example below. The EN (enable execution) is the first input and the OK (execution completed without an error) is the first output of the PAGEUDFB and the PAGE2170.

```
<table>
<thead>
<tr>
<th>PAGExxxx</th>
<th>PAGExxxx</th>
</tr>
</thead>
<tbody>
<tr>
<td>EN</td>
<td>OK</td>
</tr>
</tbody>
</table>
```

**IMPORTANT**

Do not change any of the inputs or outputs of these function blocks.

The inputs and outputs of the PAGEUDFB or the PAGE2170 will be explained in the next sections. Portions of the EXAMPLE.LDO and the EXAMPLE1.LDO are sometimes mentioned or included to illustrate the logic that could be used to interface your PiPage application to your machine control application. You can refer to other portions of the EXAMPLE.LDO or EXAMPLE1.LDO for additional information not covered here. Reading the network comments in these LDOs will also be helpful.

There are some tasks that the PAGEUDFB or PAGE2170 perform automatically and other tasks that require LDO programming on your part. This is indicated in the information that follows in the column labeled *Set by*.

Some of the tasks done automatically by the PAGEUDFB or the PAGE2170 include:

- Select and display page one on the TCS on power up.
- Display static text, touch keys, page select, bar/charts, data display/entry regions on the new page when selected.
- Display all touch buttons in their state 1 form when new page is selected.
- Refresh any data defined as refresh data when values change.
  NOTE: Bar/charts are updated whenever the setpoint, compare to, and/or chart variable [in DATA(1...255)] value change.
- Display the new page when a page select region is touched.

NOTE: The application portion of the LDO is not interfaced with Static Text Regions.
### 3.3 Page Select portion of PAGEUDFB or PAGE2170

<table>
<thead>
<tr>
<th>LDO Name</th>
<th>Long name/description</th>
<th>Set by...</th>
<th>UDFB Label</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAGE(0)*</td>
<td>Active page number latch. Only one of the array of booleans is set while on a specific page. For example, when on page one, PAGE(1) will be set; when on page two, PAGE(2) will be set, etc.</td>
<td>UDFB</td>
<td>PAGE</td>
<td>BOOL</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(1...255)</td>
<td>(1...255)</td>
</tr>
<tr>
<td>LDOPGSEL</td>
<td>Select new page on ladder event.</td>
<td>LDO/UDFB</td>
<td>PSEL</td>
<td>BOOL</td>
</tr>
<tr>
<td>LDOPAGE</td>
<td>Ladder generated page to select.</td>
<td>LDO</td>
<td>PG</td>
<td>USINT</td>
</tr>
</tbody>
</table>

*Index of zero is not used. Valid numbers are 1 - 255.
Output Side of PAGEUDFB or PAGE2170

ANYPAGE  Any page selected one-shot. Will be set for one scan when a new page is selected either from a page select region or from the ladder.  
UDFB  NEPG  BOOL

LDOPGSEL  Selects new page on ladder event. Is reset by UDFB.  
LDO/UDFB  PSEL  BOOL

PAGEACTV  Holds the number of the page that is currently active. For example, if page 5 is active, PAGEACTV will equal 5.  
UDFB  PGNM  USINT

PGSELERR  Page select error  
Set for one scan if an invalid page is selected from a page select region or a ladder page select. Either the page was not included in the file build or the page number to select is 0. The ladder will display the previous page.  
UDFB  PGER  BOOL

FRSTSCAN  Relay set on the first scan. Remains on for all subsequent scans. The positive transition of this relay can be used as a first scan oneshot and can be used for initializing servos, dynamic messages, etc.  
UDFB  FRST  BOOL

NEWPAGE  Holds the number of the page about to be selected when a page select region is touched.  
UDFB  NEWP  USINT

PAGEOK  Set when any valid page is active. Reset while a new page is being displayed.  
On power up, page one is automatically selected.
When a page is selected, PAGE(X) will remain on as long as the page is active. Only one of these relays is energized at a time by the UDFB. For example, when page 3 is active, PAGE(3) will be energized. The active relay can be used to jump around logic, qualify touch buttons, etc.

Pages can be selected from Page Select Regions designed with PiCPAGE or from the LDO. If on a page with page select regions, touching the page select region causes the new page to be selected.

When selecting pages from the LDO, you set the boolean LDOPGSEL and you set the variable LDOPAGE equal to the number of the page to be selected. PAGE function block will reset LDOPGSEL after the page selection is complete. For example, to choose page 7 from the LDO, set LDOPGSEL and let LDOPAGE equal 7.

Logic similar to that shown in the EXAMPLE.LDO for Page 16 would be used. This illustrates the selection of a new page when entering data. Selecting a new page could also be based on some LDO event such as a fault occurring, a touch button or key touched, etc.

NOTE: EXAMPLE1.LDO contains everything that is found in the EXAMPLE.LDO and so when EXAMPLE.LDO is referred to throughout this section, the information also applies to EXAMPLE1.LDO.
Page Select from the LDO - Page 16 of EXAMPLE

...60...... PAGE16: Page 16 illustrates the page select on ladder event feature. If not on page 16, jump around. A menu is displayed on page 16. The operator will enter the number of the page to select. When a page number is entered, that page will be selected by this network. Enable data entry of the page number to select. When a valid page number is entered, set LDOPGSEL to go to the new page and store the page number entered in LDOPAGE.

![Diagram]

...61...... END16:

NOTE

If you try to select an invalid page number, the boolean variable PGSELRERR will be set until a valid page is selected.

This variable can be viewed in the View Variable List when running animation in PiCPro.
### 3.4 Touch Button portion of PAGEUDFB or PAGE2170

<table>
<thead>
<tr>
<th>LDO Name</th>
<th>Long name/description</th>
<th>Set by</th>
<th>UDFB Label</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TB(0)*</td>
<td>Touch button touched one-shot. Energized for one scan when a touch button is touched. For example, if touch button 2 is touched, TB(2) will be energized for one scan.</td>
<td>UDFB</td>
<td>TB</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TR(0)*</td>
<td>Touch button released one-shot. Energized for one scan when a touch button is released.</td>
<td>UDFB</td>
<td>TR</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TBS1(0)*</td>
<td>Touch button state 1 display.</td>
<td>LDO</td>
<td>TBS1</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TBS2(0)*</td>
<td>Touch button state 2 display.</td>
<td>LDO</td>
<td>TBS2</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TBS3(0)*</td>
<td>Touch button state 3 display.</td>
<td>LDO</td>
<td>TBS3</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TBS4(0)*</td>
<td>Touch button state 4 display.</td>
<td>LDO</td>
<td>TBS4</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TBS1P(0)*</td>
<td>Touch button state 1 display on previous scan.</td>
<td>UDFB</td>
<td>TB1P</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TBS2P(0)*</td>
<td>Touch button state 2 display on previous scan.</td>
<td>UDFB</td>
<td>TB2P</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TBS3P(0)*</td>
<td>Touch button state 3 display on previous scan.</td>
<td>UDFB</td>
<td>TB3P</td>
<td>BOOL (1..40)</td>
</tr>
<tr>
<td>TBS4P(0)*</td>
<td>Touch button state 4 display on previous scan.</td>
<td>UDFB</td>
<td>TB4P</td>
<td>BOOL (1..40)</td>
</tr>
</tbody>
</table>

*Index of zero is not used. Valid numbers are 1 - 40.

**Only available in the PAGE2170 function block.
### Output Side of PAGEUDFB or PAGE2170

<table>
<thead>
<tr>
<th>LDO Name</th>
<th>Long name/description</th>
<th>Set by...</th>
<th>UDFB Label</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANYTB</td>
<td>Any touch button touched one-shot. Energized for one scan when any touch button is touched.</td>
<td>UDFB</td>
<td>NETB</td>
<td>BOOL</td>
</tr>
<tr>
<td>TBNUM</td>
<td>Number of touch button touched. For example, if touch button 6 is touched, TBNUM will equal 6 and ANYTB will be set for one scan.</td>
<td>UDFB</td>
<td>TBNM</td>
<td>USINT</td>
</tr>
</tbody>
</table>

When a touch button (TB) is touched, the PAGEUDFB or the PAGE2170 indicates to the application LDO that TB(X) was touched. It is set for one scan. (Note that TB(0) is not used.)

When a touch button (TR) is released, the PAGE2170 indicates to the application LDO that TR(X) was released. It is set for one scan.

Touch buttons can appear in from one to four different states as you define them in PiCPage. When changing to any of four defined states, always use a set coil for the new state and a reset coil for the current state. The previous scan coils (TBSXP) can be used to tell what the current state of the touch button is.

For example, to change from state 1 to state 2 for touch button 1, set TBS2(1) and reset TBS1(1). Use the previous scan coil, TBSXP(1), to tell what the current state of the touch button is.

Page 3 from the EXAMPLE.LDO is an example of how to display the four touch button states.
Touch Button - Page 3 of EXAMPLE

Page 3 is the touch button page. If not on page 3 jump around page 3 logic.

Up to 40 touch buttons can be defined per page. The touch button contacts are re-used for every page.

When a touch button is touched, the contact TB(X) will be set for one scan.

When a new page is selected, all touch buttons will be set to state 1.
Arrays of booleans are reserved to change the touch button states.
TBS1[1..40] - touch button state 1
TBS2[1..40] - touch button state 2
TBS3[1..40] - touch button state 3
TBS4[1..40] - touch button state 4
To change the display state of a touch button, set the coil for the new state and reset the coil for the state that was active.

Arrays of booleans are reserved to show the previous state of the touch buttons.
TBS1P[1..40] - touch button previously state 1
TBS2P[1..40] - touch button previously state 2
TBS3P[1..40] - touch button previously state 3
TBS4P[1..40] - touch button previously state 4

--- Page 3 ---

--- Page 3 ---

--- Page 3 ---

--- Page 3 ---

--- Page 3 ---

--- Page 3 ---
Output Side of PAGE2170 function block

<table>
<thead>
<tr>
<th>LDO Name</th>
<th>Long name/description</th>
<th>Set by...</th>
<th>UDFB Label</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCREENOK</td>
<td>The PAGE2170 function block continually monitors the communication line between the PiC and the touch screen. This Boolean is set when the PiC and the touch screen are communicating. It is reset whenever communication is lost.</td>
<td>UDFB</td>
<td>TSOK</td>
<td>BOOL</td>
</tr>
</tbody>
</table>

The networks below from page 18 of the EXAMPLE1.LDO illustrate how communication between the TCS and your ladder can be monitored.

**Touch Button Released - Page 18 of EXAMPLE1**

...69......
If the color bar is not running and the touch button is pressed, start the color bar.

```
<table>
<thead>
<tr>
<th>TAGO</th>
<th>TB(21)</th>
<th>TAGO</th>
</tr>
</thead>
</table>
```

...70......
If the color bar is running and the touch button is released, stop the color bar. Also, if we lost communication with the touch screen, stop the color bar.

Note that if an application depends on touch button release to stop a process, it should also monitor SCREENOK (PAGE2170's TSOK output) to be certain the ladder is still communicating with the touch screen. If communication is lost, the ladder will never see the touch button release.

```
<table>
<thead>
<tr>
<th>TAGO</th>
<th>TR(21)</th>
<th>TAGO</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCREENOK</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**WARNING**

If your application is relying on the release of a touch button to control a process, you must monitor the TSOK output of PAGE2170. This allows you to monitor communication between your ladder and the TCS. If communication is lost, the ladder will not recognize that a touch button has been released. Network 70 above illustrates the logic required to insure you will recognize a loss of communication.
### 3.5 Data Entry and Display portion of PAGEUDFB or PAGE2170

<table>
<thead>
<tr>
<th>LDO Name</th>
<th>Long name/description</th>
<th>Set by...</th>
<th>UDFB Label</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA(0)*</td>
<td>Data variables for entry and display. 255 data variables available from the array of for data display or entry from the application ladder or the data entry feature of PiCPage.</td>
<td>LDO/UDFB</td>
<td>DATA</td>
<td>DINT (1..255)</td>
</tr>
<tr>
<td>TBENTRY</td>
<td>Data entry by touch button. Set in the application ladder when data entry by touch button is defined in PiCPage. Data entry begins when the appropriate touch button is touched as defined in PiCPage.</td>
<td>LDO/UDFB</td>
<td>DETB</td>
<td>BOOL</td>
</tr>
</tbody>
</table>
| REGENTRY | Data entry by region number. Boolean and variable (REGNUM) used together when initiating data entry by region number. To enable data entry by region number, REGENTRY is set by the application ladder and REGNUM is set equal to the number of the data display region you would like to edit.  
NOTE: Does not require any special setup in PiCPage.                                                                 | LDO/UDFB | DERN       | USINT      |
| NLENTRY  | Data entry by next/last/right/left. Boolean and variable (REGNUM) used together when initiating next/last/right/left data entry. To enable next/last/right/left data entry NLENTRY is set by the application ladder and REGNUM is set equal to the data display type number where entry should begin. Next/last/right/left data entry is defined in PiCPage.  
The next/last/right/left touch keys also have to be defined. 
This method of data entry allows the operator to arrow through the various data entry items.                                                | LDO/UDFB | DENL       | BOOL       |
| ABRTENT  | Abort data entry. Set from the ladder to abort data entry at any time.                                                                                                                                               | LDO/UDFB | DEAB       | BOOL       |
REGNUM  Region number to enter. This is the number that appears in the Type # column of PiCPPage.

HEX(0)* Display data in hex format.

BIN(0)* Display data in binary format.

*LIndex of zero is not used. Valid numbers are 1 - 255 for DATA and 1 - 40 for HEX and BIN.

Output Side of PAGEUDFB or PAGE2170

VALIDENT  Valid entry made one-shot. The <Enter> key was pressed and the number entered is within the limits specified in PiCPPage.

INVLDENT  Invalid entry made one-shot. The <Enter> key was pressed and the number entered is not within the limits specified in PiCPPage.

ENTALWD  Data entry enabled but no keys have been touched yet. Set until data entry is made.

ENTACTV  Data entry is active and one key has been pressed. Set until data entry is made.

TBENTRY  TBENTRY is used as an output so that it can be reset.

REGENTRY  REGENTRY is used as an output so that it can be reset.

NENTRY   NENTRY is used as an output so that it can be reset.

ABRTENT  ABRTENT is used as an output so that it can be reset.

BUILTUP  Data entry buffer holding the value of the data as it is being entered.

ACTVREG  The data region number currently being edited.
Variables for data display and entry are from the array of DINTs in DATA(1...255). The index into the DATA array is specified when a data display region is created with PICPage.

Pages used for data entry must have touch keys to enter values. Keys 0 through 9 and ENT are required. Data entry can be triggered in three ways. Examples of each are shown in the EXAMPLE.LDO. Only page 4 is included here.

1. **Touch Button** [Page 4 (shown below) and 13 of EXAMPLE]

Data entry can be started after a touch button on the page is touched if this has been defined in PICPage. In the LDO, TBENTRY must be set when on the page where data entry should occur. The UDFB will reset TBENTRY.

---

**Data Entry by Touch Button - Page 4 of EXAMPLE**

...10.... PAGE4:

Page 4 is the Data Entry by touch button page.
If not on page 4 jump around page 4 logic.

On this page data entry is started by touching a touch button.

For this method of data entry, when the data entry/display regions are created with PICPage, there must be a 'yes' in entry via button and the touch button number to initiate entry must be filled in.

\[\text{PAGE}(4)\]

\[\mid / \mid \rightarrow \text{END4}\]

When page 4 is first selected, set the data entry by touch button request TBENTRY [touch button entry desired]. When a touch button is touched, it will automatically activate data entry based on the data entry/display region definition.

Touch buttons 1 and 2 are the buttons that enable data entry. Touch buttons 3 and 4 are concealed in state 1, and in state 2 display the entry limits for data 1 and data 2.

When touch button 2 is touched, and data entry has been activated ENTAILWD, set state 2 of button 2 TBS2(2) and reset state 1 of button 2 TBS1(2). This reverse videos the button. Setting state 2 of button 3 displays the limits for entry.

When an entry is made VALID, or an invalid entry is made INVALID, then clear state 2 of button 2 and set state 1 of button 2. This normal video the button. Setting state 1 of button 3 conceals the limits.
If touch button 2 is pressed, and data entry is activated, then reverse video tb2 by setting it to display state 2.

Display the limits for entry by setting display state 2 of touch button 4.

When an entry is made, normal video the button and conceal the limits by setting state 1 of buttons 2 and 4.
2. **Next/Last/Right/Left Keys** (Page 8 of EXAMPLE)

When there are several pieces of data to be entered on a page, define touch keys next, last, right, and left (i.e., ←, ↑, →, ↓) to select the item to enter. To start data entry by Next/Last/Right/Left, set NLENTRY and let REGNUM equal the region number where data entry should start.

The region number refers to the number that appears in the Type # column of the page in PiCPage as shown below.

```
Workstation Files Update rce Region Display Set state

Region # = Region Type = Type #

Region 10  Data display  #1
Region 11  Data display  #2
Region 12  Data display  #3
Region 13  Data display  #4
```

For example if you want to enter data beginning with data display region number 1, set NLENTRY and let REGNUM = 1.

It is also necessary to define in the data display entry field of PiCPage the next, last, right, and left positions by region number for each.

For example, look at the top half of page 8 on the TCS shown below. The information in the square brackets show the position moved to when the identified key is hit.

```
<table>
<thead>
<tr>
<th>Page 8 - Data Entry 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>D(100)= +0 D(101)= ←[→Right] +0 D(102)= ←[←Left] +0</td>
</tr>
<tr>
<td>D(103)= [↑Next] +0 D(104)= +0 D(105)= +0</td>
</tr>
<tr>
<td>D(106)= +0 D(107)= +0 D(108)= +0</td>
</tr>
<tr>
<td>D(109)= +0 D(110)= +0 D(111)= +0</td>
</tr>
<tr>
<td>D(112)= [↑Last] +0 D(113)= +0 D(114)= +0</td>
</tr>
</tbody>
</table>
```
Using the first data display region D(100) (shown in the dark box) as an example, open file 99 page 8 in PiCPage on your workstation. Move the cursor to Region 10 Data display #1 as shown below.

<table>
<thead>
<tr>
<th>Workstation</th>
<th>Files</th>
<th>Update tcs</th>
<th>Region</th>
<th>Display</th>
<th>Set state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region #</td>
<td>Region Type</td>
<td>Type #</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 1</td>
<td>Static text</td>
<td>#1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 2</td>
<td>Page select</td>
<td>#1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 13</td>
<td>Data display</td>
<td>#1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 11</td>
<td>Data display</td>
<td>#2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 12</td>
<td>Data display</td>
<td>#3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 13</td>
<td>Data display</td>
<td>#4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 14</td>
<td>Data display</td>
<td>#5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 15</td>
<td>Data display</td>
<td>#6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 16</td>
<td>Data display</td>
<td>#7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 17</td>
<td>Data display</td>
<td>#8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 18</td>
<td>Data display</td>
<td>#9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 19</td>
<td>Data display</td>
<td>#10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 20</td>
<td>Data display</td>
<td>#11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 21</td>
<td>Data display</td>
<td>#12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 22</td>
<td>Data display</td>
<td>#13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 23</td>
<td>Data display</td>
<td>#14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 24</td>
<td>Data display</td>
<td>#15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 40</td>
<td>Touch key</td>
<td>#1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 41</td>
<td>Touch key</td>
<td>#2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 42</td>
<td>Touch key</td>
<td>#3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Region 43</td>
<td>Touch key</td>
<td>#4</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Press <Enter>.

The data display entry field appears as shown below. A ‘Yes’ has been entered in the Next/last select field. The Type #s of the region to move to from the D(100) variable are entered for each key.

---

**Display Set state**

| Page: 8 | Region: 10 | Type: Data display #1 |

Locate region: 
corner line: 3
corner column: 1

Box attributes: Yes
Outline region: No
text: 3
size, line(s): 1
size, column(s): 18
style: 1

Text attributes: No
offset line(s): 0
offset column(s): 0
Text: 'D(100)'= 'D(100)'

Data display: 
data variable #: 100
# of digits: 3
display sign: Yes
digits to right: 0
refresh: No
double height: No
offset line(s): 0
offset column(s): 8

Data entry limits: 
upper: 999
lower: -999

Test data: 999

Entry via buffer: No
corner line: 0
corner column: 0
Next/last select: Yes
next: 4
last: 13
right: 2
left: 3

Touch button select: No
touch button: 0

Will move to this data variable when key is pressed:

- D(103)
- D(112)
- D(101)
- D(102)

---

**Interfacing the Application LDO to Pages**
The table below summarizes how the keys are defined for D(100). This was done for each of the 15 data entry regions on Page 8.

<table>
<thead>
<tr>
<th>Key</th>
<th>Page 8 Key</th>
<th>Type #</th>
<th>Page 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next</td>
<td>↓</td>
<td>move to</td>
<td>4</td>
</tr>
<tr>
<td>Last</td>
<td>↑</td>
<td>move to</td>
<td>13</td>
</tr>
<tr>
<td>Right</td>
<td>→</td>
<td>move to</td>
<td>2</td>
</tr>
<tr>
<td>Left</td>
<td>←</td>
<td>move to</td>
<td>3</td>
</tr>
</tbody>
</table>

3. **Region Number** (Page 5 of EXAMPLE)

When data entry is started from some event in the ladder, set REGENTRY and let REGNUM equal the region number where data entry should start. Region number refers to the number that appears in the Type # column of the page in PiCPage as shown above.

Data display regions can be displayed as decimal, binary, or hex values. The default is decimal. To display hex values, set variable HEX(1..40) for region number 1 to 40 with a set coil. To display binary values, set variable BIN(1..40) for region number 1 to 40 with a set coil. Reset both to return to decimal display.

When you make a valid data entry, VALIDENT is set by PAGEUDFB or PAGE2170. A valid entry is made when <ENTER> is pressed and the value is within the limits defined in PiCPage.

When you make an invalid data entry, INVLDENT is set by PAGEUDFB or PAGE2170. Invalid data means <ENTER> is pressed and the value is not within the limits defined in PiCPage.

For example, if data display number 4 should be displayed in binary form, set BIN(4). Page 14 of EXAMPLE.LDO illustrates hex/binary display and is shown below.

Data entry is done with decimal values only.

There is a refresh data option in the data display field which will update data whenever it changes if ‘Yes’ is entered. This allows you to display such things as axis position. When data entry is allowed, the value is automatically refreshed when it is entered even if ‘No’ is entered here.
Hex/Binary - Page 14 of EXAMPLE

Page 14 is the hex/binary display page.

To display data in a different format than decimal, the array of booleans BIN(1..40) and HEX(1..40) are reserved.

Data entry/display regions are numbered 1 to 40 on each page.

To display data in binary instead of decimal, set BIN(region #) when on that page.

To display data in hex instead of decimal, set HEX(region #) when on that page.

This page displays the same data variable three different ways.

The touch buttons 1 through 16 are used to set the bits in the data variable. TB(1) is bit 1, TB(2) is bit 2, etc..

<table>
<thead>
<tr>
<th>PAGE(14)</th>
<th>BIN(2)</th>
<th>HEX(3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( )</td>
<td>( )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PAGE(14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ]</td>
</tr>
</tbody>
</table>

When a button 1 through 16 is touched, reverse its state from whatever the

<table>
<thead>
<tr>
<th>TB(1)</th>
<th>TBS1P(1)</th>
<th>TBS2(1)</th>
<th>TBS1(1)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(S)</td>
<td>(R)</td>
<td></td>
</tr>
<tr>
<td>TBS2P(1)</td>
<td>TBS1(1)</td>
<td>TBS2(1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(S)</td>
<td>(R)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TB(2)</th>
<th>TBS1P(2)</th>
<th>TBS2(2)</th>
<th>TBS1(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(S)</td>
<td>(R)</td>
<td></td>
</tr>
<tr>
<td>TBS2P(2)</td>
<td>TBS1(2)</td>
<td>TBS2(2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(S)</td>
<td>(R)</td>
<td></td>
</tr>
</tbody>
</table>

[repeated through TB(16)]
Convert the touch button states (on or off depending on state 2 of touch button) to the low byte and high byte variables.

Form a 32 bit variable from the low and high byte by shifting the high byte left by 8 bits, and or-ing with the low byte. Convert from a bitwise variable to a numeric variable DATA(60) which is the variable displayed on this page in the 3 formats.

END14:
3.6 Touch Key portion of PAGEUDFB or PAGE2170

![Diagram of Touch Key portion]

<table>
<thead>
<tr>
<th>LDO name</th>
<th>Long name/description</th>
<th>Set by...</th>
<th>UDFB Label</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Side of PAGEUDFB or PAGE2170</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TK(0)</td>
<td>Touch key touched one-shot. Energizes for one scan when a touch key is touched. For example, if touch key 3 is touched TK3 will be energized for one scan. NOTE: TK(0) is the zero key, TK(1) is the one key, etc.</td>
<td>UDFB TK</td>
<td>BOOL</td>
<td>(0..39)</td>
</tr>
<tr>
<td>Output Side of PAGEUDFB or PAGE2170</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ANYTK</td>
<td>Any touch key touched one-shot. Energized for one scan when any of the 40 touch keys is touched.</td>
<td>UDFB NETK</td>
<td>BOOL</td>
<td></td>
</tr>
<tr>
<td>KEYNUM</td>
<td>Number of touch key touched. For example, if touch key 5 was touched, TKNM would equal 5 and ANYTK would be energized for one scan.</td>
<td>UDFB TKNM</td>
<td>USINT</td>
<td></td>
</tr>
</tbody>
</table>
When a touch key is touched, the PAGEUDFB or PAGE2170 indicates to the application LDO that TK(X) was touched. It is set for one scan. (Note that TK(0) is used.)

The table below shows the index number of the 40 touch keys on the left and the numeric value or function of TK(0 - 20) assigned by the PAGEUDFB or PAGE2170 when entering data on the right. TK(21 - 39) are not used by the PAGEUDFB or PAGE2170 but can be used in the application LDO.

<table>
<thead>
<tr>
<th>tk[ ] (index)</th>
<th>key</th>
<th>tk[ ] (index)</th>
<th>key</th>
<th>tk[ ] (index)</th>
<th>key</th>
<th>tk[ ] (index)</th>
<th>key</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>'0'</td>
<td>10</td>
<td>','</td>
<td>20</td>
<td>'LEFT'</td>
<td>30</td>
<td>tk[30]</td>
</tr>
<tr>
<td>1</td>
<td>'1'</td>
<td>11</td>
<td>'+'</td>
<td>21</td>
<td>tk[21]</td>
<td>31</td>
<td>tk[31]</td>
</tr>
<tr>
<td>2</td>
<td>'2'</td>
<td>12</td>
<td>'-'</td>
<td>22</td>
<td>tk[22]</td>
<td>32</td>
<td>tk[32]</td>
</tr>
<tr>
<td>3</td>
<td>'3'</td>
<td>13</td>
<td>'±'</td>
<td>23</td>
<td>tk[23]</td>
<td>33</td>
<td>tk[33]</td>
</tr>
<tr>
<td>4</td>
<td>'4'</td>
<td>14</td>
<td>'ENT'</td>
<td>24</td>
<td>tk[24]</td>
<td>34</td>
<td>tk[34]</td>
</tr>
<tr>
<td>6</td>
<td>'6'</td>
<td>16</td>
<td>'DEL'</td>
<td>26</td>
<td>tk[26]</td>
<td>36</td>
<td>tk[36]</td>
</tr>
<tr>
<td>7</td>
<td>'7'</td>
<td>17</td>
<td>'NEXT'</td>
<td>27</td>
<td>tk[27]</td>
<td>37</td>
<td>tk[37]</td>
</tr>
<tr>
<td>8</td>
<td>'8'</td>
<td>18</td>
<td>'LAST'</td>
<td>28</td>
<td>tk[28]</td>
<td>38</td>
<td>tk[38]</td>
</tr>
<tr>
<td>9</td>
<td>'9'</td>
<td>19</td>
<td>'RIGHT'</td>
<td>29</td>
<td>tk[29]</td>
<td>39</td>
<td>tk[39]</td>
</tr>
</tbody>
</table>

Touch key pads are used in pages 4, 5, 8, and 13 in the EXAMPLE.LDO for data entry. Whenever touch keys are used, touch keys 0 through 9 and the ENT key must be included.

Touch keys can also be used to build an alphanumeric key pad on the TCS to enter strings into string variable. Pages 9 and 10 are examples of this in the EXAMPLE.LDO.
### 3.7 File Display portion of PAGEUDFB or PAGE2170

<table>
<thead>
<tr>
<th>LDO name</th>
<th>Long name/description</th>
<th>Set by...</th>
<th>UDFB Label</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input Side of PAGEUDFB or PAGE2170</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISPFILE</td>
<td>RAMDISK file display request. The boolean DISPFILE and string FILENAME are used together to display an ASCII file from the RAMDISK. The boolean DISPFILE is set by the application ladder and the string FILENAME is set equal to the name of the RAMDISK filename.</td>
<td>LDO</td>
<td>FILE</td>
<td>BOOL</td>
</tr>
<tr>
<td>FILENAME</td>
<td>Filename of RAMDISK file to display. The string FILENAME must be of the following form:</td>
<td>LDO</td>
<td>NAME</td>
<td>STRING</td>
</tr>
<tr>
<td></td>
<td>'RAMDISK:file.ext$00'</td>
<td></td>
<td></td>
<td>(40)</td>
</tr>
<tr>
<td></td>
<td>The cursor positioning must be embedded in the file in the following form:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'&lt;ESC&gt;[row#;column#H'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Output Side of PAGEUDFB or PAGE2170</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISPFILE</td>
<td>DISPFILE is used as an output so that it can be reset.</td>
<td>LDO</td>
<td>FILE</td>
<td>BOOL</td>
</tr>
<tr>
<td>OPENFAIL</td>
<td>ASCII file open failed. The OPENFAIL is set if a request to display a nonexistent RAMDISK file is made.</td>
<td>UDFB</td>
<td>FAIL</td>
<td>BOOL</td>
</tr>
<tr>
<td>OPENERR</td>
<td>ASCII file open error number</td>
<td>UDFB</td>
<td>ERNM</td>
<td>INT</td>
</tr>
</tbody>
</table>
If you have created an ASCII file that contains text you want to display on one of your pages, load the file on to the RAMDISK. To display it, set DISPFILE with a set coil and put the file name in the string FILENAME. The filename string must be in the following form:

RAMDISK: file.ext $00

RAMDISK must be in all capital letters and the null ($00) is required at the end of the string. The cursor positioning must be embedded in the file. To cursor position to row (r) and column (c), use the following:

<Esc>[r;cH

where r must be from 1 to 24 and c must be from 1 to 80.

NOTE

If the ASCII file does not exist, the boolean variable OPENFAIL will be set until a valid file is requested.

This variable can be viewed in the View Variable List when running animation in PiCPro.

This feature can be used to display help screens. Page 11 from the EXAMPLE.LDO is an example of this.
### Dynamic Messages portion of PAGEUDFB or PAGE2170

<table>
<thead>
<tr>
<th>LDO name</th>
<th>Long name/description</th>
<th>Set by...</th>
<th>UDFB Label</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMSG(0)*</td>
<td>Dynamic message display request. Set by you in the application ladder when you want to display dynamic message number X. For example, to display dynamic message number 345, the boolean DMSG(345) is set by you in the ladder. When the dynamic message has been displayed, the UDFB will reset DMSG(345).</td>
<td>LDO</td>
<td>DMSG</td>
<td>BOOL (1..600)</td>
</tr>
<tr>
<td>INSRTTXT</td>
<td>Dynamic message text insert request. This boolean and variable (INSRTSTR) can be used together to append a text string to the dynamic message. When the dynamic message number X is about to display [DACK(X) is energized], then set the boolean INSRTTXT and set the INSRTSTR equal to the text to be appended.</td>
<td>LDO</td>
<td>INST</td>
<td>BOOL</td>
</tr>
<tr>
<td>INSRTSTR</td>
<td>Dynamic message text to insert. The text in INSRTSTR will be appended to dynamic message number X.</td>
<td>LDO</td>
<td>STRG</td>
<td>STRING (100)</td>
</tr>
<tr>
<td>DISPAT</td>
<td>Dynamic message location override request. The boolean at DISPAT and the variables at ROW and COL can be used to override the preset cursor position of a dynamic message. When the dynamic message is about to display [DACK(X) is energized] then set the boolean at DISPAT and set the ROW and COL equal to the desired row number and column number respectively. Dynamic message number X will be displayed at the row and column specified.</td>
<td>LDO</td>
<td>DMAT</td>
<td>BOOL</td>
</tr>
<tr>
<td>ROW</td>
<td>Row to display dynamic message at.</td>
<td>LDO</td>
<td>ROW</td>
<td>USINT</td>
</tr>
<tr>
<td>COL</td>
<td>Column to display dynamic messages at.</td>
<td>LDO</td>
<td>COL</td>
<td>USINT</td>
</tr>
</tbody>
</table>
INTMSG(0)* Initialize a group of dynamic messages (1 through 6). Use this boolean and the variable MSGPG together. LDO INIT BOOL (1..6)

MSGPG(0)* Holds the number for the page containing a group of dynamic message. This page number is defined in PiCPage and assigned to MSGPG by you in the ladder. For example, if page number 254 in PiCPage will hold dynamic messages 101 to 200, then you would move 254 into MSGPG(2) with the PiCPro MOVE function. LDO DMPG USINT (1..6)

MSGOK(0)* Dynamic message group initialization ok. UDFB DMOK BOOL (1..6)

DACK(0)* Dynamic message acknowledge. Energized by the UDFB on the scan before dynamic message number X is to be displayed. For example, on the scan before dynamic message number 203 is going to display, DACK(203) will be energized. This boolean can be used to set INSRTTXT when inserting text into the dynamic message or to set DISP when altering the cursor position of the dynamic message. UDFB DMAC BOOL (1..600)

*Index of zero is not used. Valid numbers are 1 to 600 for DMSG and DACK, 1 to 6 for INTMSG, MSGPG, MSGOK.
Output Side of PAGEUDFB or PAGE2170

INSRTTXT  INSRTTXT is used as an output so that it can be reset.  LDO INST BOOL

INVLDMSG  Invalid dynamic message requested.  UDFB DMER BOOL
The message was requested and the page of
dynamic messages was not initialized or the
requested message was not found.

DISPAT   DISPAT is used as an output so that it can be reset.  LDO DMAT BOOL
Up to six groups of dynamic messages with 100 messages per group can be
created. On the positive transition of FRSTSCAN, the six groups of dynamic
messages can be initialized by using set coils to set INTMSG(X). Messages are
displayed using the array DMSG(1..600). To display message X, set DMSG(X)
with a set coil.

The networks from the EXAMPLE.LDO that follow illustrate initialization,
inserting string messages, and changing the location of the message display via
the LDO.

Message numbers 1 to 100 are for page (or group) 1 (MSGPG1).
Message numbers 101 to 200 are for page (or group) 2 (MSGPG2).
Message numbers 201 to 300 are for page (or group) 3 (MSGPG3).
Message numbers 301 to 400 are for page (or group) 4 (MSGPG4).
Message numbers 401 to 500 are for page (or group) 5 (MSGPG5).
Message numbers 501 to 600 are for page (or group) 6 (MSGPG6).

NOTE
You cannot skip groups. The groups must be initialized in order. For example, if
you need two groups, you must use groups 1 and 2; not groups 1 and 6.
MSGOK(1) through MSGOK(6) will be set if a group has been initialized
correctly.
Dynamic Messages from the EXAMPLE
Initialize

...66......

On the first scan, initialize the dynamic messages.
Only six pages of dynamic text messages can be initialized.
It is recommended that six pages be reserved for dynamic messages only.
In this example, only 3 pages of dynamic text are used.
The first page is 253, it is for messages 1 to 100.
The second page is 254, it is for messages 101 to 200.
The third page is 255, it is for messages 201 to 300.

This network initializes three pages of dynamic messages. The conditional
contact MUST be a positive transition of FRSTSCAN.

Place the number of the first page of dynamic messages (messages 1 to 100)
in MSGPG(1) and set INTMSG(1) to initialize the first page.

Place the number of the second page of dynamic messages (101 to 200) in
MSGPG(2) and set INTMSG(2) to initialize the second page.

Place the number of the third page of dynamic messages (201 to 300) in
MSGPG(3) and set INTMSG(3) to initialize the third page.

Up to six pages of dynamic messages can be initialized. Only three are shown
here. To initialize pages 4 through 6, use the same conditional contact and
variables MSGPG(4..6) for the page numbers, and INTMSG(4..6), for the coils
to set.

When a group has initialized ok, MSGOK(1..6) will be set.

<table>
<thead>
<tr>
<th>FRSTSCAN</th>
<th>MOVE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[P]</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>EN</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
<tr>
<td>253</td>
<td>IN1</td>
</tr>
<tr>
<td></td>
<td>OUT1</td>
</tr>
<tr>
<td>254</td>
<td></td>
</tr>
<tr>
<td>255</td>
<td></td>
</tr>
</tbody>
</table>

APPLICATION NOTE 1

If you have used the box attribute with your text, do not use the DISPAT feature
to relocate text. Only the box, not the text, will be relocated to the new row and
column.

APPLICATION NOTE 2

There must always be at least one character entered in the text field of the DMSG
in PiCPage when you are working with the INSRTTXT coil and the INSRTSTR
variable in your LDO. That character can be a space if no actual text is desired.
Dynamic Messages from the EXAMPLE
Inserting a string and displaying at a different location

...65......
This network illustrates how to insert text into a dynamic message.

The time is displayed on pages 6 and 7.

Every minute (TIMECHG set), or when page 6 or page 7 is first activated, and page 6 or page 7 is active, dynamic message 201 is displayed by setting DMSG(201). The message is just a dummy message. To insert text into a dynamic message, set the dynamic message coil as usual. When DACK(201) is energized, then put the string to insert into the dynamic message into INSRTXT and set the coil INSRTXT. In this case, the current time TIMESTR (time with the seconds deleted out of the string) is inserted into the dynamic message.

On page 7 the time is displayed at a different location than programmed in the dynamic message. To display at a different location, set DISPAT and store the new row and column in ROW and COL.

If page 6 or page 7 is not active, the time is still put into TIMESTR every minute. This string is used to flag faults for the fault history page.
NOTE

If you try to display a dynamic text message that does not exist, the boolean variable INVLDMSG will be set until a valid dynamic message is requested.

This variable can be viewed in the View Variable List when running animation in PiCPro.

3.9 TCS Keyboard portion of PAGEUDFB or PAGE2170

![Diagram of KEKB and KEYHIT connections]

<table>
<thead>
<tr>
<th>LDO name</th>
<th>Long name/description</th>
<th>Set by...</th>
<th>UDFB</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYSTYPD</td>
<td>String that holds the characters from the optional keyboard or TCS information other than touch reports.</td>
<td>UDFB</td>
<td>CHRS</td>
<td>STRING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(100)</td>
</tr>
</tbody>
</table>

Output Side of PAGE function block

KEYHIT  Any information other than a touch report coming from the TCS oneshot. For example, if the external keyboard is connected to the touch screen, when the characters are typed, the boolean KEYHIT will be set for one scan. When it is set, the string KEYSTYPD will hold the characters that have been typed on the keyboard.

When information from the TCS is something other than a touch report, that information is held in the KEYSTYPD string. This information could come from the optional keyboard connected to the TCS, a bar code reader, etc.

KEYHIT will be set for one scan when information other than touch reports is sent from the TCS.
Index

A
A099 example 2- 47
A099.BN1 1- 13
ABK file 2- 2
AP BN file 2- 2
AP file 2- 2
application
maximum 1 - 1
number 2- 13
application note
dynamic messages 3- 29
array 2- 35, 39
ASCII file 2- 2
ASCII
copy file to tcs 2- 2
create file 2- 18
attributes
box 2- 25
define 2- 2
dynamic text message length 2- 43
text 2- 27
auto-flash
touch key 2- 31

B
bar
define 2- 38
bar/chart
define 2- 5
region 2- 37
variable 38
BN file 2- 17
buffer
data display 2- 37
build file 2- 17

C
cables
pin outs 1- 9, 12
required 1- 4
cell 2- 30
chart 2- 7
master 2- 29, 30
on TCS 2- 6
characters
double height 2- 46
double width 2- 46
extended 2- 45
special set 2- 45
chart variable number 2- 39
close file 2- 17
color list 2- 26
COM1 port 2- 2
combining
UDFB/LDO 3- 1
compare to variable 2- 40
example 2- 42
computer
setup 2- 10
computer
requirements 1- 4
conceal
color 2- 25
region 2- 25
connect
TCS to PiC900 1- 12
TCS to workstation 1- 9
workstation to PiC900 1- 12
copy
ASCII file to TCS 2- 2, 18
region 2- 20
cursor
toggle 2- 23
cursor mode 2- 27

D
data display
buffer area 2- 37
digits 2- 35
digits to right 2- 35
eexamples 2- 36
hex/binary 3- 19
example 3- 20
location 2- 36
refresh 2- 35
sign 2- 35
text size 2- 36
variable 2- 35
data display/entry 2- 34, 3- 13
define 2- 5
data entry
by n/l/r/l keys 3- 17
by region number 3- 19
by touch button 3- 15
limits 2- 36
location 2- 37
next/last 2- 37
touch button 2- 37
example 3- 15
delete
region 2- 20
design pages 1- 14, 2- 1
directories 1- 8
display
menu 2- 21, 23
dynamic messages 3- 26
application note 3-29
change location example 3-30
groups 3-28
initialize example 3-29
insert string example 3-30
dynamic text
define 2-4
display 2-22
message length 2-43
region 2-42

E
edit
region 2-19
error
build 2-17
file already exists 2-13
hex characters 2-28
no TCS connected 2-18
sending file to RAMDISK 1-13
updating TCS 1-11
EXAMPLE.LDO 1-13
extended character set 2-44
extensions 2-2

F
file
ASC 2-18
BN 2-17
LST 2-18
file display 3-24
example 3-25
filename 2-1
files
ABK 2-2
AP 2-2
ASC 2-2
BN 2-2
LST 2-2
menu 1-10, 2-13, 16
save 2-2
find
region 2-20
full scale
  bar/chart 2-39

G
generate file 2-18

H
hardware
  requirements 1-4
help <F9> 2-43
hexadecimal
    characters 2-28
messages 2-44

I
insert
  region 2-19
installing
    software 1-7

K
key
  touch 2-4

L
LDO
  run 1-15
  linking
    UDFB/LDO 3-1
    LST file 2-2, 18

M
manuals 1-4
master cell 2-29, 30
memory
  TCS 2-18
menu
  display 2-21, 23
  files 1-10, 2-13, 16
  levels 2-8
  main 2-9
  region 2-19
  set state 2-21, 23
  update tcs 2-18
  workstation 2-9
merge 1-14
messages
  length for dynamic text 2-43
  length for touch button 2-32
  special 2-44
move
  region 2-20

N
new file
  create 2-13
next/last
  data entry 2-37

O
offset
  column 2-27, 38
  line 2-27, 38
open file 1-10, 2-14
outline region 2-26
  size 2-26
P
page
maximum 1- 1
number 2- 13
page select 3- 5
define 2- 5
example 3- 7
from LDO 3- 6
from PiCPage 3- 6
region 2- 33
PAGE2170 1- 3
PAGEUDFB 1- 2
automatic tasks 3- 4
use 1- 14
PiC900 1- 4
PiCPage
define 1- 1
design pages 2- 1
files 1- 7
load 1- 7
procedure overview 1- 5
run 1- 10
software 2- 1
PiCPro
run 1- 13
pin outs 1- 9, 12
port
COM1 2- 2
print file 2- 18
printer
setup 2- 11
R
RAMDISK 1- 13, 2- 2
refresh
data 2- 35, 3- 4, 19
region
attributes 2- 2, 25
bar/chart 2- 5, 37
copy 2- 20
create 2- 19
data display/entry 2- 5, 34
define 1- 1
delete 2- 20
dynamic text display 2- 4
edit 2- 19
find 2- 20
insert 2- 19
location 2- 6, 29
maximum 2- 2
menu 2- 19
move 2- 20
outline 2- 26
page select 2- 5, 33
shape 2- 6
size 2- 6, 29
static text 2- 24
static text display 2- 4
touch button 2- 5
touch key 2- 4, 29
S
save file 2- 16
screen
copyright 1- 10
set state
menu 2- 21, 23
setpoint variable 2- 39
example 2- 41
simulate
page on workstation 2- 21
test data 2- 36
software
requirements 1- 4
start up 1- 6
static text
define 2- 4
region 2- 24
T
TCS
clear 2- 21
display page 2- 21
display region 2- 21
keyboard 3- 31
memory 2- 18
model 1- 4
parameters 1- 6
setup 1- 6
test data
display 2- 22
simulate 2- 23
text
attributes 2- 27, 28
double height 2- 36, 46
double width 2- 46
dynamic 2- 42
hexadecimal characters 2- 28
length 2- 28
static 2- 24
touch area 2- 29, 30
touch button 3- 8
activate state 2- 22, 32
define 2- 5
example 3- 10
region 2- 31
state 2- 32, 3- 9
touch button released 3- 12
touch key 2- 4, 3- 22
auto flash 2- 31
examples 3- 31
list 2- 30
region 2-29
table of values 3-23
touch screen (2170)
monitoring 3-12

U
UDFB/LDO
linking 3-1
update tcs menu 2-18
user defined function block 1-1

V
variable
chart number 2-39
compare 2-40

setpoint 2-39
viewing
states 2-22, 32

W
workstation 1-4
menu 2-9
simulate page display 2-21

Z
zero scale
bar/chart 2-39