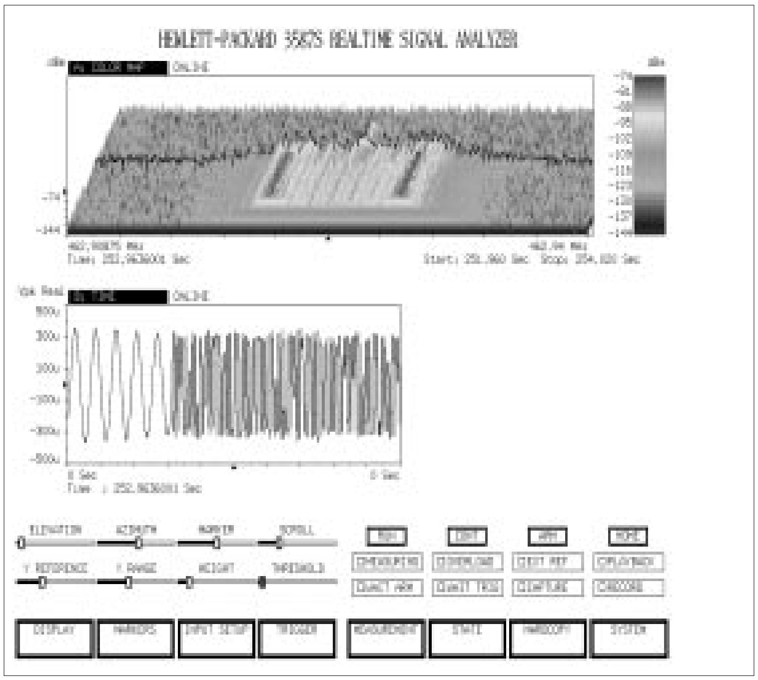
HP 35687B Option AGG Customization and Programming Reference




```
/****************************************************************************
* demod.c
*
* This file implements AM, FM, and PM demodulation along with PHASE via the
* display function customization functionality of the HP3587s software.
*
*****************************************************************************/
#include <stdlib.h>
#include <math.h>
#include "dispType.h"
#include "header.h"
#include "demod.h"

/****************************************************************************
* Add the following lines to the end of the 'aDispTypeManager' structure and
* increment the display type count by 7 in the file 'dispType.c'.
*
"PHASE",
   DATA_TYPE_TIME, phase , GRAPH_TYPE_TRACE, 6, HELP_PHASE,
   NULL, NULL, NULL, NULL, NULL, NULL, &yPhaseRange,

"FM",
   DATA_TYPE_TIME, fm, GRAPH_TYPE_TRACE, 7, HELP_FM,
   NULL, NULL, NULL, NULL, NULL, NULL, &yFMRange,
"FM_STRIP",
   DATA_TYPE_TIME, fm, GRAPH_TYPE_STRIP_CHART, 7, HELP_FM_STRIP,
   NULL, NULL, NULL, NULL, NULL, NULL, &yFMRange,

"FM_MAP",
   DATA_TYPE_TIME, fm, GRAPH_TYPE_MAP, 7, HELP_FM_MAP,
   NULL, NULL, NULL, NULL, NULL, NULL, &yFMRange,

"AM",
   DATA_TYPE_TIME, am, GRAPH_TYPE_TRACE, 8, HELP_AM,
   NULL, NULL, NULL, NULL, NULL, NULL, &yAMRange,

"AM_STRIP",
   DATA_TYPE_TIME, am, GRAPH_TYPE_STRIP_CHART, 8, HELP_AM_STRIP,
   NULL, NULL, NULL, NULL, NULL, NULL, &yAMRange,

"PM",
   DATA_TYPE_TIME, pm, GRAPH_TYPE_TRACE, 9, HELP_PM,
   NULL, NULL, NULL, NULL, NULL, NULL, &yPMRange,
*****************************************************************************/

/****************************************************************************
* The scaling structures are formatted as shown below. For a complete
* definition of each of the fields, see the dispType.h include file.
*
*  struct aScaleDesc <name> = {
*      <yRefMode>, <yRangeMode>, <maxYref>, <minYref>, <defaultYref>,
*      <defaultYRangeIndex>, <upperLimit>, <lowerLimit>, <noneCoordLabel>,
*      <yRange[ 0 -  5]>,
*      <yRange[ 6 - 11]>,
*      <yRange[12 - 17]>,
*      <yRange[18 - 23]>
*  };
*
*****************************************************************************/
struct aScaleDesc yPhaseRange = {
      Y_REF_MODE_CENTER, Y_RANGE_LIN, 180.0, -180.0, 0.0,
```

HEWLETT PACKARD

HP Part Number 35687-90027
Software Version: A.03.04 and B.03.04

Printed in U.S.A.
Print Date: May 1997

8600 Soper Hill Road Everett, Washington 98205-1298 U.S.A.

HEWLETT-PACKARD COMPANY

3000 Hanover St.

Palo Alto, CA 94304

Printing History

Edition 1: December 1995

Table of Contents

In This Book

This book introduces you to the HP 35687B Option AGG Customization Software. When you first install the HP 35687B Option AGG software, the knob box, button box, keyboard and menu system are all given default settings. You may find it useful to change these settings to fit your needs. The information in this book will teach you how to use the HP 35687B Option AGG Customization Software to do the following.

- Starting the Real Time Analyzer from an HP-UX shell.
- Create and design your own customization for the knob box, button box and keyboard accelerators.
- Change the menu system
- Develop macro programs
- Create your own display colors and color maps
- Make print data compatible with other printers using your own or a commercially available print data conversion program
- Customize markers and display functions by writing sub-routines in C
- Access captured time data
- Control the Real Time Analyzer with another program via the command port
- Command port

The tasks are written with the assumption that you are familiar with the HP-UX operating system and the X Windows system. You should have a working knowledge of basic file manipulation functions and experience using an HP-UX text editor such as *vi*. The “Developing your own functions” chapter is intended for experienced C language programmers. You also need to have some experience in operating the HP 3587S Realtime Signal Analyzer.

Note

The A.03.04 and B.03.04 versions of the HP35687B software are functionally equivalent. A.03.04 runs under HP-UX 9.0x and B.03.04 runs under HP-UX 10.20. The filesystem layout is significantly different in HP-UX 10.20. To enable transparent use of the A.03.04 configuration files when running B.03.04, the /usr/hp3587s directory is symbolically linked to the /opt/hp3587s directory by the installation process. For more detail, see the HP 35687B Installation Note.

Conventions

The following describes the general syntax rules used in this book.

- Comma (,) – separates the label name from the command. For example azimuth, AZIMUTH
- Pound Sign (#) – lets you add comments to your customization file. The comment can have its own line or can be placed at the end of a line with a command. Place a pound sign at the beginning of each line of comment text as follows:

  ```
  # comment text
  ```

- Whitespace – separates parameters from commands. For example the space between DISP_TYPE and SPECTRUM in the command

  ```
  DISP_TYPE  SPECTRUM
  ```

  is mandatory. A space inside a command/function name, such as,

  ```
  THRES HOLD
  ```

  is not allowed. A space immediately following a comma is ignored. For example `azimuth,  AZIMUTH`.
- [SOFTKEY] – represents a softkey (F1 through F8).
- Colon (:) – separates the descriptor from the command field.

1

Startup Options and System Configuration

Use this chapter to learn about the startup options and the system configuration.

Starting the program

If the computer prompts you to login after turning it on, perform the following:

1 Login

At the login prompt, enter your user name and password.

2 Start HP3587s

In a window command line, type: **`hp3587s <Return>`**

See the following page for command line options.

Command line options

To display the command-line options list, type: **`hp3587s -u <Return>.`** The current value of each option is displayed in the far right bracketed field.

Command Options

Switch <Parameters>	Description	Default Setting
-a <address>	Logical Address of E1485A	[128]
-b <buttonFile>	Button Box Description File	[BUTTONSYSTEM]
-c <colorFile>	Color Description File	[COLORSYSTEM]
-d <display>	Display	[ ]
-f <font>	Font	[10x20]
-e	Don't use Button Box	
-g <downloadable>	Downloadable Filename	[SPECTRUM]
-h	Use VXI hardware	
-j <menuFile>	Menu System File	[MENUSYSTEM]
-k	Uncouple Center Freq and Span	[Coupled]
-l	List current setup	
-m	Function Key Menus Active	
-n	Don't use Knob Box	
-p <printFile>	Printer Configuration File	[PRINTSYSTEM]
-r <accelFile>	Keyboard Accelerator File	[ACCELSYSTEM]
-s <spos>	Signal Processor Operating System	[spos]
-t <size>	Window Size 0:small 1:med 2:large	[LARGE]
-u	Display this text	
-v <macroFile>	Macro Description File	[MACROS]
-w <name>	Name for X window	[HP3587S]
-x <position>	Window position	[1]
-y <position>	Window position	[1]
-z <knobFile>	Knob Description File	[KNOBSYSTEM]
-A <mode>	Beeper, 0:Off 1:On	[On]
-B	Monochrome Display Colors	
-C <codeword>	Security Codeword (16 characters)	[ ]
-E <downFile>	Down Converter Description File	[DOWNCONVERTER]
-F <font>	Title Font	[12x24]
-G <codeword>	Option ATR Security Codeword	[ ]
-I <address>	Logical Address of E1430A/E1437A	[129]
-K <sensitivity>	Knob Sensitivity	[10]
-L <mode>	Local Bus Mode, 0:Off 1:On	[On]
-M <bytes>	Maximum Memory for Data	[16000000]
-N <lif volume>	E1562 LIF Volume Name (Opt ATR)	[default]
-O	Don't Display Opening Screen	
-S <directory>	System Directory	[/usr/hp3587s]
-T <address>	Logical Address of E1562 (Opt ATR)	[0]
-W <directory>	Working Directory	[./]
-X <mode>	Command Port, 0:Off 1:On	[On]
-Y <key>	Command Port Shared Ram Key	[0x3587]
-Z <interface>	VXI Interface Name	[vxi]

Example

To run the hp3587s analyzer with a small window (640x480) and have the X-windows border displayed, you would type the following:

```
hp3587s -t0 -wExample
```

Configuration Information

The command line is used for infrequent changes to the analyzer configuration. A more permanent change is made by editing the CONFIGURATION file. Shown below is the default configuration file, named CONFIGURATION.

CONFIGURATION file listing

```
# $Header: CONFIGURATION,v 1.6 1996/12/19 21:00:28 hmgr Exp $
#####################################
#  HP3587S REALTIME SIGNAL ANALYZER #
#        CONFIGURATION FILE         #
#####################################

### USER INTERFACE ######
Menu System Active     : 1
Knob Box Active        : 1
Button Box Active      : 1
Window Size            : 2     # 0:Small  1:Medium  2:Large
Position X             : 1
Position Y             : 1
Window Name            : HP3587S
Opening Screen         : 1
Knob Sensitivity       : 10     # 1: most sensitive  500:least sensitive
Error Messages         : 1
Warning Messages       : 1
Status Messages        : 1
Command Port           : 1
Command Port Key       : 0x3587
### MEMORY USAGE ########
Max Data Size          : 30000000    # 0:Limit of machine  0:#bytes
### FILE SYSTEM #########
Working Directory      : ./
System Directory       : /usr/hp3587s/
### CUSTOMIZATION FILES #
Printer Filename       : PRINTSYSTEM
Down Conv Filename     : DOWNCONVERTER
Button Box Filename    : BUTTONSYSTEM
Color Map Filename     : COLORSYSTEM
Menu System Filename   : MENUSYSTEM
Accelerator Filename   : ACCELSYSTEM
Macro Filename         : MACROS
Knob Box Filename      : KNOBSYSTEM
### HARDWARE SETUP ######
e1485a Logical Address : 128
e143Xa Logical Address : 129
VXI Interface          : vxi
SPU Opsys              : spos
SPU Filename           : SPECTRUM
Use Hardware           : 1
Local Bus Enabled      : 1
Center Frequency / Span: 1
### SECURITY ############
```

When the program runs, it searches for the CONFIGURATION file first in the current directory (`./`). If it doesn't find it in the current directory, it looks in /usr/hp3587s/ directory. If it's not present in either directory, the program uses internal defaults as described in the previous usage listing. The following table describes the elements in the CONFIGURATION file.

Configuration Definitions

Element Name	Value/Type	Value's Effect
Menu System Active	0:	The function key menu is turned off.
	1:	The function key menu is present.
Knob Box Active	0:	The Knob Box is not used.
	1:	The Knob Box is used.
Button Box Active	0:	The Button Box is not used.
	1:	The Button Box is used.
Window Size	0:	Small window 640x480 pixels
	1:	Medium window 1024x768 pixels
	2:	Large window 1280x1024 pixels
Position X	n:	Specifies the left location of the window.
Position Y	n:	Specifies the top location of the window.
Window Name	string:	Specifies the name of the window used by the analyzer.
Opening Screen	0:	No opening screen graphics displayed.
	1:	Opening screen graphics are displayed.
Knob Sensitivity	n:	Specifies the knob box sensitivity. 1 is the most sensitive and 500 is the least sensitive.
Error Messages	0:	Error messages are not displayed.
	1:	Error messages are displayed.
Warning Messages	0:	Warning messages are not displayed.
	1:	Warning messages are displayed.
Status Messages	0:	Status messages are not displayed.
	1:	Status messages are displayed.
Command Port	0:	The command port is disabled.
	1:	The command port is enabled.
Command Port Key	n:	Specifies command port shared RAM key. (See “Command Port Section”).
Max Data Size	n:	Specifies the amount of RAM to be used for data storage. If 0 bytes are specified
Working Directory	directory:	Specifies the directory to store data
System Directory	directory:	Specifies the directory where all the default system customization files are stored. Be sure to use the last slash as shown in the CONFIGURATION listing shown earlier.
Printer Filename	filename:	Specifies the name of the printer customization file.
Down Conv Filename	filename:	Specifies the name of the down converter customization file.
Button Box Filename	filename:	Specifies the name of the button box customization file.
Color Map Filename	filename:	Specifies the name of the color map customization file.
Menu System Filename	filename:	Specifies the name of the menu system customization file.
Accelorater Filename	filename:	Specifies the name of the keyboard accelerator customization file.
Macro Filename	filename:	Specifies the name of the macro customization file.
Knob Box Filename	filename:	Specifies the name of the knob box customization file.
e1485a Logical Address	n:	Specifies the VXI logical address of the e1485a DSP module.
e143Xa Logical Address	n:	Specifies the VXI logical address of the e1430a/e1437a ADC module.
VXI Interface	string:	Specifies VXI interface session name.
SPU Opsys	filename:	Specifies the filename of the e1485a operating system.
SPU Filename	filename:	Specifies the filename of the e1485a executable.
Use Hardware	0:	Don't use the VXI hardware.
	1:	Use the VXI hardware.
Local Bus Enabled	0:	Don't use the VXI local bus for fast data transfers.
	1:	Use the VXI local bus for fast data transfers.
Center Frequency / Span	0:	Don't couple the center frequency and span parameters.
	1:	Couple the center frequency and span parameters.

2

Customization Overview

Use this chapter to learn the general steps for making customizations for the knob box, button box, keyboard accelerators, menu system, color system, macros, and printing system. Chapters 2 through 10 to discuss specific instructions for each kind of customization.

Customization at a Glance

The HP35687B Option AGG Customization Software lets you optimize the HP 3587S Realtime Signal Analyzer for your needs. You can customize the functionality of the knob box, button box, menu system, use of color, downconverter usage, and keyboard accelerators by modifying ASCII files. You can also create your own functions and macros, and make the HP 3587S compatible with your printer. The software runs in HP-UX using X Windows and it drives the VXI-based hardware for the HP 3587S.

When you want to customize one of the customization files, you can make most customizations by remembering this three-step procedure:

1. Copy
2. Edit
3. Activate

Copy

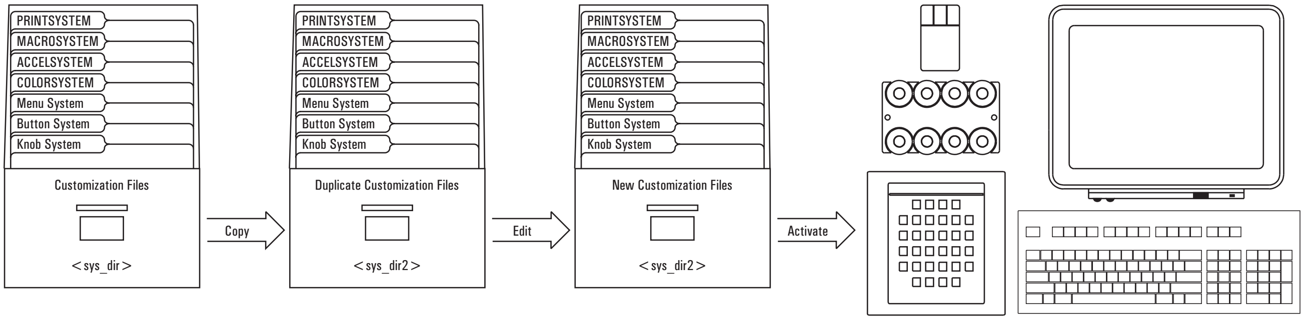
Copy the appropriate default customization file from your system directory to your work directory. See “Copying a customization file to the work directory” for more information.

Edit

Edit the customization file by editing the default information in the customization files. Each customization file consists of groups of descriptors with label names and their associated commands. Each group has a group number, a group label, and a list of commands to be assigned to each knob, button, or key. Refer to the appropriate chapter which includes a task that describes how to edit the associated customization file. Also, refer to “Printing the Programming Reference” in chapter 2 to get a hardcopy listing of the commands from which you can choose.

Activate

Make the change active for the current session by running the HP 35687B software or, if it is already running, return to the analyzer window. Then refer to the appropriate chapter which includes a task that describes how to activate each kind of customization file.



Copy the existing custom files to another (working) directory, modify the copies, then activate the customization files in the working directory.

Copying a Customization File to the Work Directory

1 Open a window on your system

2 Type:

```
cp  <sys_dir>/<customize_files>  <work_dir>/<customize_files>
```

For example:

```
cp  /usr/hp3587s/KNOBSYSTEM  $HOME
cp  /usr/hp3587s/BUTTONSYSTEM  $HOME
cp  /usr/hp3587s/ACCELSYSTEM  $HOME
cp  /usr/hp3587s/MENUSYSTEM  $HOME
cp  /usr/hp3587s/COLORSYSTEM  $HOME
cp  /usr/hp3587s/MACROSYSTEM  $HOME
cp  /usr/hp3587s/PRINTSYSTEM  $HOME
cp  /usr/hp3587s/DOWNCONVERTER  $HOME
```

3 Refer to the appropriate chapter to learn how to modify each type of customization file and save the changes.

The default customization files can be found under your system directory (<sys_dir>), which is /usr/hp3587s at installation . Keep the default files in the <sys_dir> and make a copy of them in your working directory (<work_dir>). The location of the working directory is set in the CONFIGURATION file.

Note It is important that you not keep your modified files in the /usr/hp3587s directory. Future releases of the software will overwrite these files and changes will be lost.

Printing the Programming Reference

To program macros, the knob box, the button box, the menu system, keyboard accelerator, and the command port, programming commands are used. You can print a listing of these commands with these steps:

1 Go to the HP 3587S window and make sure your printer is connected properly, turned on and ready to print.

2 Print the program reference:

[HOME]
[HARDCOPY]
[PRINT PRGM MANUAL]

3 You should get a printout like the one shown below.

If you have problems printing, refer to “To print data to a printer” in the *HP 35687B Operator's Reference* for information on printing to an HP LaserJet printer. If you have a different kind of printer, refer to the “Printing to other types of printers” chapter in this book.

The following is an example of the Programming Reference printout.

```
         HP3587S Programming Reference [B.03.04]


accel_group   [ ]Knob  [ ]Button  [X]Menu  [ ]Accel  [ ]Macro  [ ]Prog Port
 Syntax: accel_group  x   where: [0 <= x <= 6]
 Query : accel_group?

     Assigns a specific key on the keyboard to a function.  This is
     convenient for performing a certain operation from a different menu
     than the one in which it appears.  There are some default keyboard
     accelerators.  For example, `P` key to prints the full screen to the
     currently specified printer.  A list of these default keyboard
     accelerators appear in Appendix B of the HP 35687A Operator's
     Reference.  If you have software option AGG, you can create your own
     keyboard accelerators.  See the HP 35687A Option AGG Customization
     Guide to learn how.


active_trace  [ ]Knob  [X]Button  [X]Menu  [X]Accel  [X]Macro  [X]Prog Port
 Syntax: active_trace [a | b | c | ab | bc | ac | all]
 Query : active_trace?

     Specifies which trace (or set of traces) to which a parameter will be
     applied.  This is useful for applying one parameter to one trace and
     a different parameter to another trace.  If multiple traces are
     active (e.g., ABC), the same parameter change applies to each of the
     active traces if it is appropriate.
```

3

Customizing the Knob Box

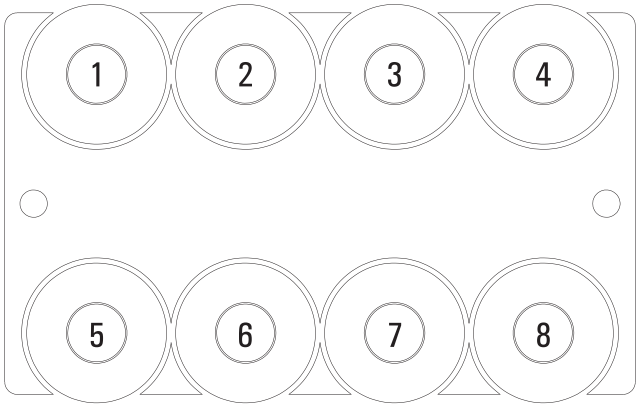
This chapter shows how to customize the knob box functions.

Introduction

The HP 35687B Option AGG uses several customization files to define the user interface. At installation, these files are located in the system directory (<sys_dir>). The default directory for <sys_dir> is /usr/hp3587s/.

The knob box customization file is named KNOBSYSTEM. The customization file associates nine user-defined commands and corresponding labels with each of the knobs. Each set of nine knob definitions is called a *group* and each group has a number corresponding to the function key numbers at the top of the system keyboard. A maximum of seven groups may be defined. Groups 1, 2, and 3 are given default definitions as defined in appendix A of the *Operator's Reference*. Knob function titles appear above the corresponding slide bars on the display.

A group definition is selected with the main keyboard's function keys or buttons on the button box. The default knob box definitions shown on the knob box template are represented by three colors; a red group, a blue group and a green group. The button box buttons that select which of the definitions are active are the leftmost three in the lower row and are color-coded. In the default KNOBSYSTEM file, groups 4, 5, and 6 are the same as groups 1, 2, and 3, respectively, and may be changed to meet your needs.



The knob-number system used to assign functions to knobs.

Default KNOBSYSTEM file

```
Group    :4
Title    :EXAMPLE #1
Knob  1 :ELEVATION,          elevation
Knob  2 :AZIMUTH,            azimuth
Knob  3 :MARKER,             move_marker
Knob  4 :SCROLL,             scroll_index
Knob  5 :Y REFERENCE,        y_ref
Knob  6 :Y RANGE,            y_range
Knob  7 :HEIGHT,             trace_height
Knob  8 :THRESHOLD,          threshold

Group    :5
Title    :EXAMPLE #2
Knob  1 :SPAN FREQ,          span_freq
Knob  2 :CENTER FREQ,        center_freq
Knob  3 :RESOLUTION,         resolution
Knob  4 :INPUT RANGE,        input_range
Knob  5 :Y REFERENCE,        y_ref
Knob  6 :Y RANGE,            y_range
Knob  7 :X REFERENCE,        x_reference
Knob  8 :X MAGNIFY,          x_magnify

Group    :6
Title    :EXAMPLE #3
Knob  1 :COLORMAP,           colormap
Knob  2 :# COLORS,           num_colors
Knob  3 :TRACE COLOR,        line_color
Knob  4 :GRID COLOR,         grid_color
Knob  5 :BACKGND COLOR,      back_color
Knob  6 :LOG FACTOR,         clr_map_log
Knob  7 :X BREAKPOINT,       clr_map_linx
Knob  8 :Y BREAKPOINT,       clr_map_liny
```

KNOBSYSTEM Conventions

- Files are organized by groups of commands.
- Group titles and program commands are not case sensitive.
- Label names are displayed exactly as entered by the user. They are limited to a maximum of 12 characters including any white spaces.
- Group 1, 2, and 3 are internally defined as the system defaults. These defaults may be superseded by defining new 1, 2, and 3 groups in a customization file.
- The maximum number of groups that can be assigned is seven.
- Only commands identified in the Command Reference with the knob entry box checked can be used. See “Printing the Programming Reference” in chapter 2 to obtain a hardcopy listing of all commands.
- Only one command can be used per line.

The keywords are Group, Title, and Knob (number). The Group keyword defines the group number and the Title keyword defines the group title that appears as a menu selection under [SYSTEM], [CUSTOMIZATION], [KNOB BOX].

If any knobs in a group are not assigned, they are assigned a blank label and a nop command.

Editing the KNOBSYSTEM file

Before you start, make sure you complete the steps in “Copying a customization file to the work directory.”

1 From your <work_dir>, bring a copy of the KNOBSYSTEM file into your editor. It should appear as shown earlier in the default file listing.

If you want to start with an original file, copy the KNOBSYSTEM file from the <sys_dir> to the <usr_dir>.

2 Enter the group number and title using the Group and Title keywords. Using group numbers 1, 2, or 3 makes that group part of the system default. Group numbers 4 through 7 must be activated by the user after the HP 35687B software is running.

3 Replace current commands with the new commands. When you choose a new command, choose from those which appear in the command dictionary reference with the “KNOB” entry box checked. (“Printing the Programming Reference” in chapter 2 shows how to obtain this listing.) For example, replacing the first three command lines with the following group assigns [ELEVATION], [AZIMUTH] and [TRACE HEIGHT] to knobs 1, 2, and 3. The remaining knobs have no function.

4 Save the changes you just made and return to the HP-UX command line.

Activating the new KNOBSYSTEM file

If KNOBSYTEM was modified while the HP 3587S was not running, starting the analyzer causes the customizations to become active in the new session. However, if KNOBSYSTEM was modified during an HP 3587S session, preset the customization files by completing the following steps.

1 Return to the HP 3587S window and make sure the software is running.

2 Press
[HOME]
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

3 Press
[HOME]
[SYSTEM]
[CUSTOMIZE]
[KNOB BOX]

4 Select the new knob system title from the menu. The knob box should now operate using your customized file.

You can now use the top three knobs to control elevation, azimuth, and trace height, respectively.

If your customization is not successful, an error message appears on the status line of the HP 3587S window. Make sure each command is entered exactly as shown in the command dictionary reference.

"Undoing" a KNOBSYSTEM customization

If you have simply added a group to the KNOBSYSTEM file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified KNOBSYSTEM file in the working directory. However, if you want to "start over again" by reverting back to the original KNOBSYSTEM file in the system directory, perform the following.

1 Open a window on your operating system.
2 Remove the KNOBSYSTEM file from your working directory.
3 Return to the HP 3587S window.
4 Press
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

With no KNOBSYSTEM file in the working directory, presetting the customization files causes the software to use the default KNOBSYSTEM file in the system directory.

4

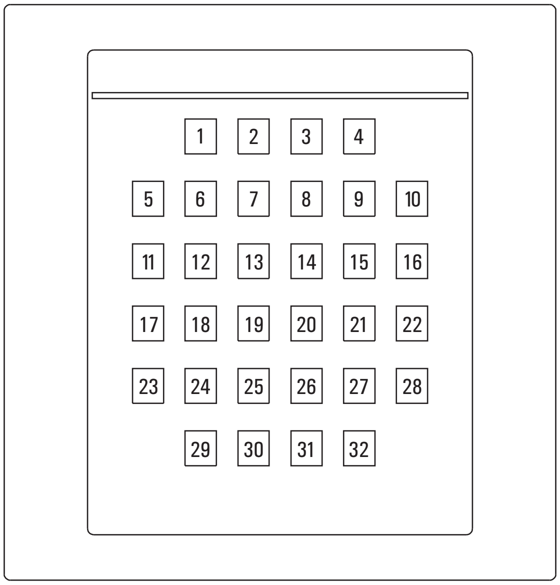
Customizing the Button Box

This chapter shows how to customize the button box.

Introduction

The customization file for the button box is called BUTTONSYSTEM. Since the button box has 32 buttons, the customization file associates 32 user defined commands with 32 labels. Each set of 32 is a group. The button box can only have one group assigned to it at a time. The default commands for these settings are shown on the following page.

The default group is group 1. The BUTTONSYSTEM file duplicates group 1 into group 2 to be modified to fit your needs.



The button-number system used to assign functions to buttons.

Default BUTTONSYSTEM file

```
Group     :2
Title     :EXAMPLE
Button  1 :RUN,                  run
Button  2 :PAUSE/CONT,           pause/cont
Button  3 :ARM,                  arm
Button  4 :HOME,                 home_menu
Button  5 :SPECTRUM,             disp_type spectrum
Button  6 :SPECTRAL MAP,         disp_type spectral_map
Button  7 :SPECTROGRAM,          disp_type spectrogram
Button  8 :ROLOGRAM,             disp_type rollogram
Button  9 :COLOR MAP,            disp_type color_map
Button 10 :TIME,                 disp_type time
Button 11 :MARKER OFF,           marker_mode off
Button 12 :SINGLE TRACE,         disp_format single
Button 13 :DUAL TRACE,           disp_format dual
Button 14 :TRIPLE TRACE,         disp_format triple
Button 15 :OVERLAY TRACE,        disp_format overlay
Button 16 :MARKER TO PEAK,       marker_func 0
Button 17 :SINGLE MARKER,        marker_mode single
Button 18 :A,                    active_trace a
Button 19 :B,                    active_trace b
Button 20 :C,                    active_trace c
Button 21 :ALL TRACES,           active_trace all
Button 22 :NEXT PEAK RGHT,       marker_func 2
Button 23 :RELATIVE MARKER,      marker_mode relative
Button 24 :HIDDEN LINE,          hidden_line toggle
Button 25 :WIRE FRAME,           wireframe toggle
Button 26 :GRID,                 grid toggle
Button 27 :SCROLL DIR,           scroll_dir toggle
Button 28 :NEXT PEAK LEFT,       marker_func 1
Button 29 :X/Y DISPLAY,          knob_group 0
Button 30 :MONITOR,              knob_group 1
Button 31 :COLOR,                knob_group 2
Button 32 :Y AUTOSCALE,          y_auto_scale
```

BUTTONSYSTEM Conventions

- Files are organized by groups of commands.
- Group titles and program commands are not case sensitive.
- Label names are displayed exactly as you enter them. They are limited to a maximum of 12 characters including any spaces.
- Group 1 is defined as the system default. Changing this group in the BUTTONSYSTEM file changes the default button box functions.
- The maximum number of groups that can be assigned is seven.
- Only commands identified in the Command Reference with the button entry box checked can be used. See “Printing the Programming Reference” in chapter 2 to obtain a hardcopy listing of all commands.
- Only one command can be entered per line.

The keywords are Group, Title, and Button (number). The Group keyword defines the group number and the Title keyword defines the group title that appears as a menu selection under [SYSTEM], [CUSTOMIZATION], [BUTTON BOX]. The remaining 32 lines identify the label names and commands that correspond to the buttons on the button box.

The default button box interface uses the first group in the button box configuration (Group 1) as the default file when the application is first brought up. Changes made to Group 1 cause the new customization to become the default.

If any buttons in a group are not assigned, they are assigned a blank label and nop command.

Editing the BUTTONSYSTEM file

Before you start, make sure you perform the steps in “Copying a customization file to the work directory” in chapter 2.

1 From your <work_dir>, bring a copy of the BUTTONSYSTEM file into your editor. It should appear as shown earlier in the default file listing.
2 Make a backup copy of this default group configuration.
3 Enter the group number and the title using the Group and Title keywords. Specifying group number 1 makes that group part of the system default. Group numbers 2 through 7 must be activated by the user after the HP 35687B software is running.
4 Replace the existing commands with new commands. Choose commands from those which appear in the command dictionary reference with the “BUTTON” entry box checked. (“Printing the Programming Reference” in chapter 2 shows how to obtain this listing.) For example, replacing the first 4 command lines with the following commands assigns [HISTOGRAM], [MENU ACTIVE], [ZOOM MODE] and [PRINT SCREEN] to buttons 1, 2, 3, and 4.
5 Save the changes you just made and return to the HP-UX command line.

Activating the new BUTTONSYSTEM file

If BUTTONSYSTEM was modified while the HP 3587S was not running, starting the analyzer will cause the customizations to become active in the new session. However, if the customizations were made during an HP 3587S session, preset the customization files by completing the following steps.

1 Return to the HP 3587S window and make sure the software is running.

2 Press
[HOME]
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

3 Press
[HOME]
[SYSTEM]
[CUSTOMIZE]
[BUTTON BOX]

4 Select the new button system label from the menu. The button box should now operate using your customized file.

You can now use the top four buttons to select the [HISTOGRAM], [MENU ACTIVE], [ZOOM MODE] and [PRINT SCREEN] functions.

If your customization is not successful, an error message appears in the status line on the HP 3587S display. Make sure each command is entered exactly as shown in the command dictionary reference.

"Undoing" a BUTTONSYSTEM customization

If you have simply added a group to the BUTTONSYSTEM file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified BUTTONSYSTEM file in the working directory. However, if you want to "start over again" by reverting back to the original BUTTONSYSTEM file in the system directory, perform the following.

1 Open a window on your operating system.
2 Remove the BUTTONSYSTEM file from your working directory.
3 Return to the HP 3587S window.
4 Press
[HOME]
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

With no BUTTONSYSTEM file in the working directory, presetting the customization files causes the software to use the default BUTTONSYSTEM file in the system directory.

5

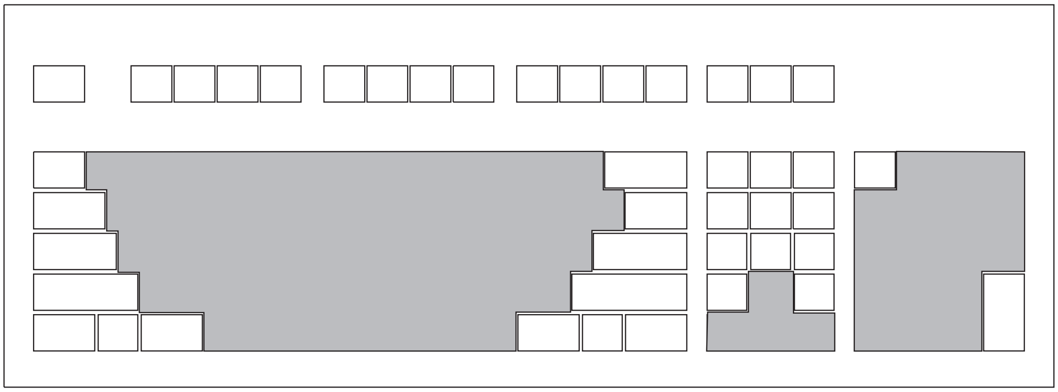
Customizing the Keyboard

This chapter shows how to customize the keyboard accelerator system.

Introduction

The customization file for the keyboard accelerator system is called ACCELSYSTEM. This file allows you to assign functions to a key on the keyboard. For example, one of the default keyboard accelerators is the “P” key which prints the full screen to the printer. Any function you use regularly can be set as a “speed key” by adding it to the ACCELSYSTEM file.

You can have up to 7 groups of keyboard accelerator systems. Group 1 is the default keyboard accelerator system. Groups 2 through 7 must be activated once the software is running. The ACCELSYSTEM customization file associates keys on the keyboard with a command.



The shaded areas indicate which keys are available to use as keyboard accelerators (both upper and lower case versions of each).

Default ACCELSYSTEM file

```
Group         : 2
Title         : EXAMPLE
RIGHT-ARROW   : step_marker 1
LEFT-ARROW    : step_marker -1
UP-ARROW      : step_scroll 1
DOWN-ARROW    : step_scroll -1
SH-RIGHT-ARROW: step_marker 10
SH-LEFT-ARROW : step_marker -10
SH-UP-ARROW   : step_scroll 10
SH-DOWN-ARROW : step_scroll -10
PAGE-DOWN     : page_scroll -1
PAGE-UP       : page_scroll 1
P             : print_screen
R             : recply_mode record
S             : sync_scales
i             : disp_i_state
d             : disp_h_state
m             : disp_m_state
s             : swap_markers
t             : disp_d_state
```

ACCELSYSTEM Conventions

- Files are organized by groups of commands.
- Group titles and program commands are not case sensitive, but key names are case sensitive. For example, “P” can be used to print the full screen to a printer and “p” can print the current instrument state to a printer.
- Only single keys can be used for keyboard accelerators. Key combinations are not allowed.
- Up to 105 keyboard accelerators can be defined within one group.
- Group 1 is defined as the system default. This default may be superseded by defining a new group 1 menu system in a customization file.
- The maximum number of groups that can be assigned is seven.
- Only commands identified in the command dictionary reference with the Accel entry box checked can be used. See “Printing the Programming Reference” in chapter 2 to obtain a hardcopy listing of all commands.
- Only one command can be entered per line.
- You should not assign commands which can change or lose data to keys that can be accidentally pressed. For example, you should not assign “p” to preset all and “P” to print_screen.
- Valid keys include the upper and lower case versions of all the shaded keys shown in the figure on an earlier page. This includes characters ASCII 31 (!) through ASCII 126 (~) and the following keywords for the others:

UP-ARROW	RIGHT-ARROW	SH-LEFT-ARROW	PAGE-UP
DOWN-ARROW	SH-UP-ARROW	SH-RIGHT-ARROW	SPACE
LEFT-ARROW	SH-DOWN-ARROW	PAGE-DOWN	COLON

The keywords are Group and Title. The Group keyword defines the group number and the Title keyword defines the group title that appears as a menu selection under [SYSTEM], [CUSTOMIZE], [KEYBOARD DEFAULT]. The remaining lines identify the key names and commands that correspond to the keys on the keyboard.

Editing the ACCELSYSTEM file

Before you start, make sure you complete the steps in “Copying a customization file to the work directory” for the ACCELSYSTEM file in chapter 2.

1 From your <work_dir>, bring a copy of the ACCELSYSTEM file into your editor.

2 Enter the group number and title using the Group and Title keywords. Specifying group number 1 makes that group part of the system default. Group numbers 2 through 7 must be activated by the user after the HP 35687B software is running. (See “Activating a customization”.)

3 Select the desired key and its associated command. Any single ASCII character is eligible to use as a keyboard accelerator. When you select a command, choose from those which appear in the command dictionary reference with the “ACCEL” entry box checked. (“To print the Programming Reference” in chapter 2 shows how to obtain this listing.) The following example defines a keyboard accelerator system which assigns all of the display types to a “speed key”.

```
Group         : 2
Title         : DISP_TYPE
a             : disp_type spectrum
b             : disp_type spectral_map
c             : disp_type spectrogram
d             : disp_type rollogram
e             : disp_type color_map
f             : disp_type pr_spectrum
g             : disp_type time
h             : disp_type time_map
i             : disp_type strip_chart
j             : disp_type pr_time
k             : disp_type histogram
l             : disp_type pdf
m             : disp_type cdf
n             : disp_type hist_map
o             : disp_type hist_cmap
p             : disp_type off
```

4 Save the changes you just made and return to the HP-UX command line.

Activating the new ACCELSYSTEM file

If ACCELSYSTEM was modified while the HP 3587S was not running, starting the analyzer causes the customizations to become active in the new session. However, if ACCELSYSTEM was modified during an HP 3587S session, preset the customization files by completing the following steps.

1 **Return to the HP 3587S window and make sure the software is running.**

2 **Press**
[HOME]
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

3 **Press**
[HOME]
[SYSTEM]
[CUSTOMIZE]
[KEYBOARD]

4 **Select the new menu system label (in the example, [DISP_TYPE]). The keyboard accelerator system should now operate using your customized file.**

If your customization is not successful, an error message appears in the status line on the HP 3587S display. Make sure each command is entered exactly as shown in the command dictionary reference.

"Undoing" an ACCELSYSTEM customization

If you have simply added a group to the ACCELSYSTEM file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified ACCELSYSTEM file in the working directory. However, if you want to "start over again" by reverting back to the original ACCELSYSTEM file in the system directory, perform the following.

1. Open a window on your operating system.
2. Remove the ACCELSYSTEM file from your working directory.
3. Return to the HP 3587S window.
4. Press
   [HOME]
   [SYSTEM]
   [PRESET]
   [CUSTOMIZE FILES]

With no ACCELSYSTEM file in the working directory, presetting the customization files causes the software to use the default ACCELSYSTEM file in the system directory.

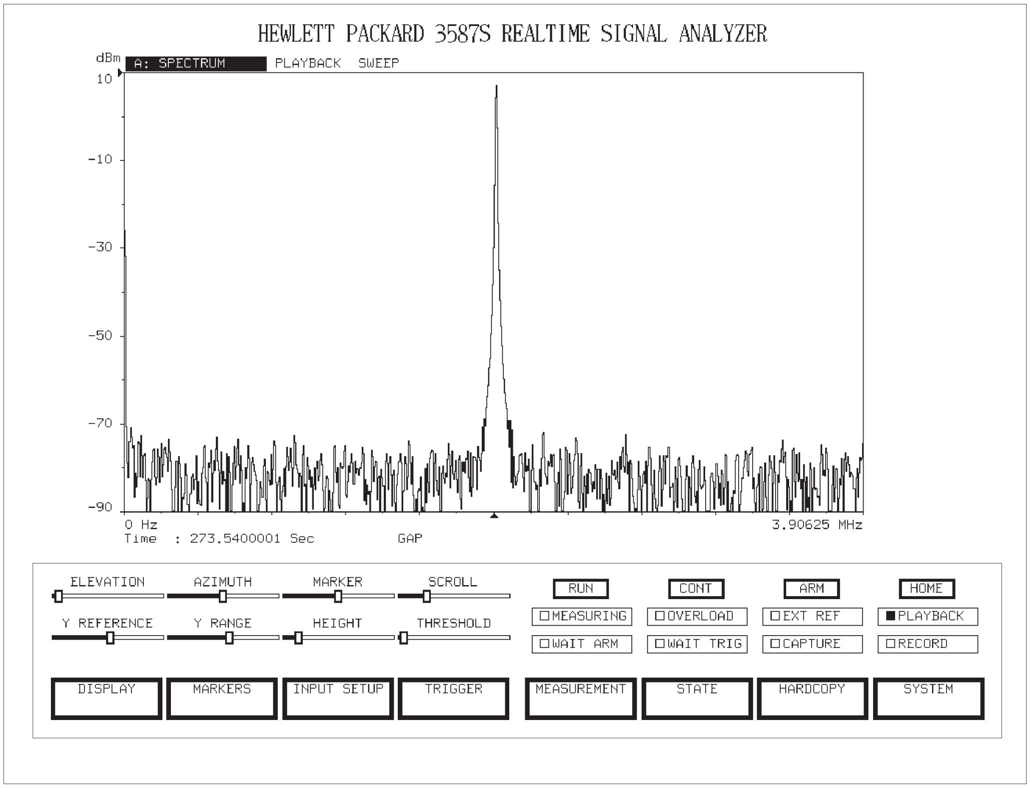
6

Customizing the Menu System

This chapter shows how to customize the menu system.

Introduction

The customization file for the menu system is called MENUSYSTEM. This is useful for defining the softkey menus to meet the specific needs of special users. For example, if you have users who only need to load a measurement state, run a spectrogram measurement and print the results, you can develop a menu system that has all these keys on the top level of the menu. Similarly, if you do not need the triggering functions, you can simplify the user interface by creating a menu system that has no trigger menus.



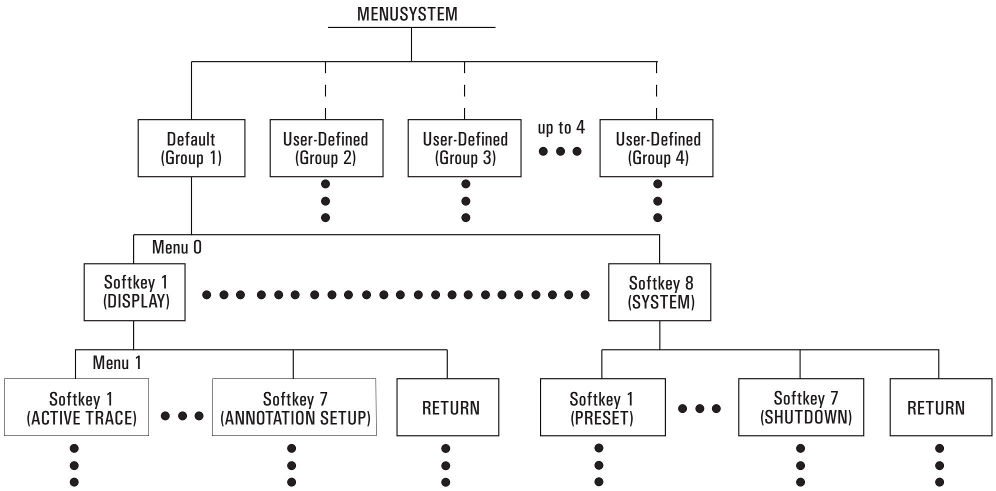
Menu entries appear in the buttons at the bottom of the screen.

You can have as many as 4 groups of menu systems. Group 1 is the default menu system. Groups 2 through 4 must be activated while the software is running. The MENUSYSTEM customization file associates each menu level with a menu number and associates each softkey label with a command. Each menu level is given a number. The following partial listing shows the default first and second-level menu.

Default MENUSYSTEM file (partial listing)*

```
Group      :2
Title      :EXAMPLE
Menu       :1
Menu Title :MAIN MENU
Menu Help  :
Key  1     :DISPLAY,,                  load_menu 2
Key  2     :MARKERS,,                  load_menu 12
Key  3     :INPUT SETUP,,              load_menu 4
Key  4     :TRIGGER,,                  load_menu 5
Key  5     :MEASUREMENT,,              load_menu 6
Key  6     :STATE,,                    load_menu 3
Key  7     :HARDCOPY,,                 load_menu 7
Key  8     :SYSTEM,,                   load_menu 8
Menu       :2
Menu Title :DISPLAY MENU
Menu Help  :
Key  1     :ACTIVE TRACE,,             active_trace
Key  2     :DSPLY TYPE,,               disp_type
Key  3     :DSPLY SETUP,,              load_menu 9
Key  4     :DSPLY FORMAT,,             disp_format
Key  5     :COORDINATES,,              load_menu 10
Key  6     :SCALE,,                    load_menu 11
Key  7     :ANNOTATION,SETUP,          load_menu 21
Key  8     :RETURN,,                   load_menu -1
Menu       :3
Menu Title :INSTRUMENT STATE MENU
Menu Help  :
Key  1     :SAVE STATE,,               save_state
Key  2     :RECALL STATE,,             recall_state
Key  3     :HARD DISK,UTILITIES,       load_menu 32
Key  4     :INPUT,STATE,               disp_i_state
Key  5     :MEASUREMENT,STATE,         disp_m_state
Key  6     :DATA HEADER,STATE,         disp_h_state
Key  7     :TRACE,STATE,               disp_d_state
Key  8     :RETURN,,                   load_menu -1
```

*This is an incomplete list of the default MENUSYSTEM. Display MENUSYSTEM in a window for a complete listing.



MENUSYSTEM Conventions

- Group titles and program commands are not case sensitive.
- Label names are displayed exactly as you enter them. You can have a two-line label. The top and bottom lines can have to 12 characters (including spaces). A comma with no spaces separates the top line from the bottom line of a label.
- Group 1 is defined as the system default. This default may be superseded by defining a new group 1 menu system in a customization file.
- The maximum number of groups that can be assigned is four.
- Only commands identified in the command dictionary reference with the "MENU" entry box checked can be used. See "Printing the Programming Reference" in chapter 2 to obtain a hardcopy listing of all commands.
- Only one command can be entered per line.
- Each menu system group should include the shutdown command so that the system can be shut down from any of the customization file groups.
- Each sub-menu should include the return command for returning to the next menu level up.
- The maximum help text length is 512 characters.

The keywords are Group, Title, and Menu (number). The Group keyword defines the group number and the Title keyword defines the group title that appears as a menu selection under [SYSTEM], [CUSTOMIZE], [MENU SYSTEM]. The third line identifies which menu number appears when the [HOME] key is pressed. The remaining lines identify the menu numbers and labels, the help text for each menu, and softkey labels and commands for each menu.

Editing the MENUSYSTEM file

Before you start, make sure you complete the steps in “Copying a customization file to the work directory” in chapter 2.

1 From your <work_dir>, bring a copy of the MENUSYSTEM file into your editor. It should appear as shown in figure below.

2 Enter the group number and create the user title. Specifying group number 1 makes that group part of the system default. Group numbers 2 through 4 must be activated by the user after the HP 35687B software is running. (See “To active a customization”.)

3 Replace current commands with the new commands. When you choose a new command, choose from those which appear in the command dictionary reference with the “MENU” entry box checked. (“Printing the Programming Reference” in chapter 2 shows how to obtain this listing.) This menu recalls a state, moves the marker to the peak signal, allows an annotation line, saves the data to a file and prints the trace data.

```
Group      :2
Title      :MENU SYS 2
Menu       :1
Menu Title :MAIN MENU
Menu Help  :
Key  1     :DATA,COLLECTION,            load_menu 2
Key  2     :MARKERS,,                   load_menu 12
Key  3     :INPUT SETUP,,               load_menu 4
Key  4     :TRIGGER,,                   load_menu 5
Key  5     :MEASUREMENT,,               load_menu 6
Key  6     :STATE,,                     load_menu 3
Key  7     :HARDCOPY,,                  load_menu 7
Key  8     :SYSTEM,,                    load_menu 8
Menu       :2
Menu Title :DATA COLLECTION MENU
Menu Help  :
Key  1     :RECALL STATE,,              recall_state
Key  2     :MARKER,FUNCTIONS,           marker_func
Key  3     :ANNOTATION,,                load_menu 21
Key  4     :SAVE DATA,,                 save_data
Key  5     :PRINT SCREEN,,              print_screen
Key  7     :SHUTDOWN,,                  load_menu 33
Key  8     :RETURN,,                    load_menu -1
```

4 Save the changes you just made and return to the HP-UX command line.

Activating the new MENUSYSTEM file

If MENUSYSTEM was modified while the HP 3587S was not running, starting the analyzer causes the customizations to become active in the new session. However, if MENUSYSTEM was modified during an HP 3587S session, preset the customization files by completing the following steps.

1 **Return to the HP 3587S window and make sure the software is running.**

2 **Press**
[HOME]
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

3 **Press**
[HOME]
[SYSTEM]
[CUSTOMIZE]
[MENU SYSTEM]

4 **Select the new menu system label (in the example, [MENU SYS 2]). The menu system should now operate using your customized file.**

When you press the [HOME] key, the new "DATA COLLECTION" selection should appear on the first softkey.

If your customization is not successful, an error message appears in the status line on the HP 3587S display. Make sure each command is entered exactly as shown in the command dictionary reference.

"Undoing" a MENUSYSTEM customization

If you have simply added a group to the MENUSYSTEM file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified MENUSYSTEM file in the working directory. However, if you want to "start over again" by reverting back to the original MENUSYSTEM file in the system directory, perform the following.

1 Open a window on your operating system.
2 Remove the MENUSYSTEM file from your working directory.
3 Return to the HP 3587S window.
4 Press
[HOME]
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

With no MENUSYSTEM file in the working directory, presetting the customization files causes the software to use the default MENUSYSTEM file in the system directory.

Note This works only if the customize command is included in MENUSYSTEM. If it is not, you must shutdown the analyzer and restart it.

Note You should *always* include the shutdown command in MENUSYSTEM.

7

Customizing the Color System

This chapter shows how to customize the color assignments for 3-D map colors.

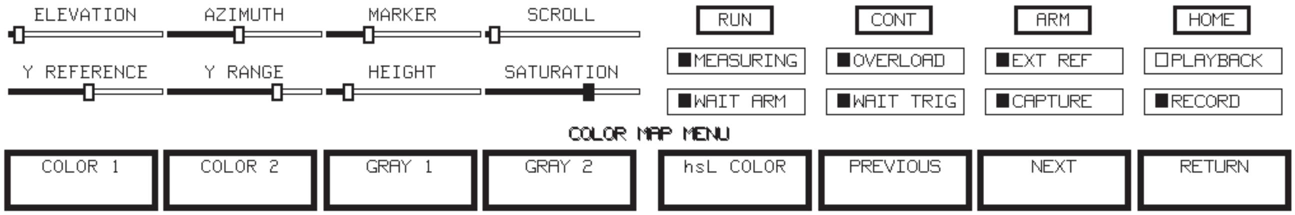
Introduction

The customization file for the color system is called COLORSYSTEM. This is useful for specifying the colors used in color maps, spectrograms, and digital persistence displays.

You can have up to 17 groups of color systems. Group 1 is the default color system. Groups 2 through 17 must be activated with the software running. Each group can take on one of three color modes; a COLOR mode, a GRAYSCALE mode and a USER mode.

The COLOR mode is based on a standard RGB (red-green-blue) color model which blends from one hue to another like a rainbow. See the section on Color Graphics in *Starbase Graphics Techniques Volume 1* (HP Part No. 98592-90080) for a picture of this color scale. With the COLOR mode, you can define a color system that uses any portion of this model. The GRAYSCALE mode is based on a standard gray scale model which blends from light shades of gray(white) to dark shades of gray(black). You can define a color system that uses any portion of this model. The USER mode lets you define your own color model. For more information on RGB color models and HSL (hue-saturation-luminance) models, refer to the section on Color Graphics in *Starbase Graphics Techniques Volume 1* .

The COLORSYSTEM file associates each color setup with the softkeys which appear when you press [DISPLAY], [DSPLY SETUP], [3-D MAP COLORS], [MAP COLOR] as shown in the following illustration.



The Color Map menu.

Default COLORSYSTEM File*

```
Group           : 1
Title           : COLOR 1
Mode            : COLOR
Start Hue       : 0.70
Stop Hue        : 0.00

Group           : 2
Title           : COLOR 2
Mode            : COLOR
Start Hue       : 0.00
Stop Hue        : 0.70

Group           : 3
Title           : GRAY 1
Mode            : GRAYSCALE
Start Intensity : 1.00
Stop Intensity  : 0.10

Group           : 4
Title           : GRAY 2
Mode            : GRAYSCALE
Start Intensity : 0.10
Stop Intensity  : 1.00

Group           : 5
Title           : hsL COLOR
Mode            : USER
Number of Colors: 16
HSL Color 16    : 0.130 1.000 1.000
HSL Color 15    : 0.130 1.000 0.9375
HSL Color 14    : 0.130 1.000 0.875
HSL Color 13    : 0.130 1.000 0.8125
HSL Color 12    : 0.130 1.000 0.750
HSL Color 11    : 0.130 1.000 0.6875
HSL Color 10    : 0.130 1.000 0.625
HSL Color 9     : 0.130 1.000 0.5625
HSL Color 8     : 0.130 1.000 0.500
HSL Color 7     : 0.130 1.000 0.4375
HSL Color 6     : 0.130 1.000 0.375
HSL Color 5     : 0.130 1.000 0.3125
HSL Color 4     : 0.130 1.000 0.250
HSL Color 3     : 0.130 1.000 0.1875
HSL Color 2     : 0.130 1.000 0.125
HSL Color 1     : 0.130 1.000 0.0625
```

```
Group           : 6
Title           : hSl COLOR
Mode            : USER
Number of Colors: 16
HSL Color 1     : 0.130 1.000   1.000
HSL Color 2     : 0.130 0.9375  1.000
HSL Color 3     : 0.130 0.875   1.000
HSL Color 4     : 0.130 0.8125  1.000
HSL Color 5     : 0.130 0.750   1.000
HSL Color 6     : 0.130 0.6875  1.000
HSL Color 7     : 0.130 0.625   1.000
HSL Color 8     : 0.130 0.5625  1.000
HSL Color 9     : 0.130 0.500   1.000
HSL Color 10    : 0.130 0.4375  1.000
HSL Color 11    : 0.130 0.375   1.000
HSL Color 12    : 0.130 0.3125  1.000
HSL Color 13    : 0.130 0.250   1.000
HSL Color 14    : 0.130 0.1875  1.000
HSL Color 15    : 0.130 0.125   1.000
HSL Color 16    : 0.130 0.0625  1.000


Group           : 7
Title           : RANDOM
Mode            : USER
Number of Colors: 68
RGB Color 1     : 50 191 193              # Aquamarine
RGB Color 2     : 0 147 143               # MediumAquamarine
RGB Color 3     : 0 0 0                   # Black
RGB Color 4     : 0 0 255                 # Blue
RGB Color 5     : 95 146 158              # CadetBlue
RGB Color 6     : 34 34 152               # CornflowerBlue
RGB Color 7     : 56 75 102               # DarkSlateBlue
RGB Color 8     : 176 226 255             # LightBlue
RGB Color 9     : 124 152 211             # LightSteelBlue
RGB Color 10    : 50 50 204               # MediumBlue
RGB Color 11    : 106 106 141             # MediumSlateBlue
RGB Color 12    : 47 47 100               # MidnightBlue
RGB Color 13    : 35 35 117               # NavyBlue
RGB Color 14    : 114 159 255             # SkyBlue
RGB Color 15    : 126 136 171             # SlateBlue
RGB Color 16    : 84 112 170              # SteelBlue
RGB Color 17    : 255 114 86              # Coral
RGB Color 18    : 0 255 255               # Cyan
RGB Color 19    : 142 35 35               # Firebrick
RGB Color 20    : 165 42 42               # Brown
RGB Color 21    : 244 164 96              # SandyBrown
RGB Color 22    : 218 170 0               # Gold
RGB Color 23    : 239 223 132             # Goldenrod
RGB Color 24    : 209 193 102             # MediumGoldenrod
RGB Color 25    : 0 255 0                 # Green
```

```
RGB Color 26     : 0 86 45                 # DarkGreen
RGB Color 27     : 85 86 47                # DarkOliveGreen
RGB Color 28     : 80 159 105              # ForestGreen
RGB Color 29     : 0 175 20                # LimeGreen
RGB Color 30     : 50 129 75               # MediumForestGreen
RGB Color 31     : 52 119 102              # MediumSeaGreen
RGB Color 32     : 35 142 35               # MediumSpringGreen
RGB Color 33     : 115 222 120             # PaleGreen
RGB Color 34     : 82 149 132              # SeaGreen
RGB Color 35     : 65 172 65               # SpringGreen
RGB Color 36     : 50 216 56               # YellowGreen
RGB Color 37     : 47 79 79                # DarkSlateGrey
RGB Color 38     : 84 84 84                # DimGrey
RGB Color 39     : 168 168 168             # LightGrey
RGB Color 40     : 126 126 126             # Gray
RGB Color 41     : 179 179 126             # Khaki
RGB Color 42     : 255 0 255               # Magenta
RGB Color 43     : 143 0 82                # Maroon
RGB Color 44     : 255 135 0               # Orange
RGB Color 45     : 239 132 239             # Orchid
RGB Color 46     : 139 32 139              # DarkOrchid
RGB Color 47     : 189 82 189              # MediumOrchid
RGB Color 48     : 255 181 197             # Pink
RGB Color 49     : 197 72 155              # Plum
RGB Color 50     : 255 0 0                 # Red
RGB Color 51     : 107 57 57               # IndianRed
RGB Color 52     : 213 32 121              # MediumVioletRed
RGB Color 53     : 255 69 0                # OrangeRed
RGB Color 54     : 243 62 150              # VioletRed
RGB Color 55     : 233 150 122             # Salmon
RGB Color 56     : 150 82 45               # Sienna
RGB Color 57     : 222 184 135             # Tan
RGB Color 58     : 216 191 216             # Thistle
RGB Color 59     : 0 0 1                   # Transparent
RGB Color 60     : 25 204 223              # Turquoise
RGB Color 61     : 0 166 166               # DarkTurquoise
RGB Color 62     : 0 210 210               # MediumTurquoise
RGB Color 63     : 156 62 206              # Violet
RGB Color 64     : 138 43 226              # BlueViolet
RGB Color 65     : 245 222 179             # Wheat
RGB Color 66     : 255 255 255             # White
RGB Color 67     : 255 255 0               # Yellow
RGB Color 68     : 173 255 47              # GreenYellow

Group            : 8
Title            : COLOR 3
Mode             : COLOR
Start Hue        : 0.25
Stop Hue         : 0.75
```

This is an incomplete list of the default COLORSYSTEM. Display COLORSYSTEM in a window for a complete listing.

COLORSYSTEM Conventions

- Group titles are not case sensitive.
- Label names are displayed exactly as entered by the user. You can have a one-line label. The line can have as many as 16 characters (including spaces).
- Group 1 is defined as the system default. Changing this group in the COLORSYSTEM file changes the default colors.
- The maximum number of groups that can be assigned is 17.
- For USER (color mode) groups, the values for RGB and HSL components must be entered using either a decimal system (from 0.000000 to 1.000000) or an integer system (from 1 to 255).

The keywords are Group, Title, and Mode. The Group keyword defines the group number and the Title keyword defines the group title that appears as a menu selection under [DISPLAY], [DSPLY SETUP], [3-D MAP COLORS], [MAP COLOR]. The Mode keyword defines whether the rest of the entries are Color parameters, Grayscale parameters, or User parameters.The remaining lines identify the parameters appropriate for that mode as outlined in the following discussion.

Keywords are:

- `Group` - defines the group number
- `Title` -defines the group title that appears as a menu selection
- `Mode` - defines how to interpret the rest of the parameters.
  The mode is either `USER`, `COLOR`, or `GRAYSCALE`.
  - If the mode is `USER`, then the parameters used are `Number Of Colors`, and either `RGB Color` or `HSL Color`.
  - If the mode is `COLOR`, then the parameters define `Start` and `Stop Hue.`
  - If the mode is `GRAYSCALE`, the parameters define the `Start` and `Stop Intensity`.
- `Number of Colors` - defines the number of user-defined colors.
- `RGB Color` - defines one color using the red-blue-green format.
- `HSL Color` - defines one color using the hue-saturation-luminosity format.
- `Start Hue` - defines one boundary in the color spectrum.
- `Stop Hue` - defines the other boundary in the color spectrum.
- `Start Intensity` - defines one boundary in the grayscale spectrum.

- `Stop Intensity` - defines the other boundary in the grayscale spectrum.

Editing the COLORSYSTEM file

Before you start, make sure you complete the steps in "Copying a customization file to the work directory".

1. From your <work_dir>, bring a copy of the COLORSYSTEM file into your editor.
2. Enter the group number and create the user title. Specifying group 1 makes that group part of the system default. Group numbers 2 through 17 must be activated by the user while the HP 35687B software is running.
3. Enter the mode; COLOR, GRAYSCALE, or USER.
4. Each mode requires entry different parameters.

a. If the mode is COLOR, enter the start hue and stop hue values. The following hue values represent a gradation between the following colors.

Hue Values	Colors
0.00 - 0.10	orange
0.10 - 0.25	yellow
0.25 - 0.40	green
0.40 - 0.50	blue-green
0.50 - 0.60	blue
0.60 - 0.75	purple
0.75 - 0.85	pink
0.85 - 1.00	red

For example, the following COLOR group represents a color system that starts with yellow, which gradates to green, blue-green and then blue.

```
Group          : 2
Title          : COLOR 2
Mode           : COLOR
Start Hue      : 0.10
Stop Hue       : 0.60
```

b. If the mode is Grayscale, enter the start intensity and stop intensity. An intensity of 1.00 is pure white and an intensity of 0.00 is pure black. A GRAYSCALE group may look like the following.

```
Group           : 3
Title           : GRAY 1
Mode            : GRAYSCALE
Start Intensity : 1.00
Stop Intensity  : 0.10

Group           : 4
Title           : GRAY 2
Mode            : GRAYSCALE
Start Intensity : 0.10
Stop Intensity  : 1.00
```

c. If the mode is User, enter the total number of colors you want to use in the color system. Then enter the red, green and blue content for each color. These entries can either have *decimal* values ranging from 0.0000000 to 1.0000000 within a group or *integer* values ranging from 1 to 255 within a group. For example, the following group shows a random selection of colors for a USER color system.

```
Group           : 11
Title           : RANDOM
Mode            : USER
Number of Colors: 16
RGB Color 1     : 50 191 193            # Aquamarine
RGB Color 2     : 0 147 143             # MediumAquamarine
RGB Color 3     : 0 0 0                 # Black
RGB Color 4     : 0 0 255               # Blue
RGB Color 5     : 95 146 158            # CadetBlue
RGB Color 6     : 34 34 152             # CornflowerBlue
RGB Color 7     : 56 75 102             # DarkSlateBlue
RGB Color 8     : 176 226 255           # LightBlue
RGB Color 9     : 124 152 211           # LightSteelBlue
RGB Color 10    : 50 50 204             # MediumBlue
RGB Color 11    : 106 106 141           # MediumSlateBlue
RGB Color 12    : 47 47 100             # MidnightBlue
RGB Color 13    : 35 35 117             # NavyBlue
RGB Color 14    : 114 159 255           # SkyBlue
RGB Color 15    : 126 136 171           # SlateBlue
RGB Color 16    : 84 112 170            # SteelBlue
```

5 Save the changes you just made and return to the HP-UX command line.

Activating the new COLORSYSTEM file

If COLORSYSTEM was modified while the HP 3587S was not running, starting the analyzer will cause the customizations to become active in the new session. However, if COLORSYSTEM was modified during an HP 3587S session, preset the color system by completing the following steps.

1 Return to the HP 3587S window and make sure the software is running.

2 Press
[HOME]
[SYSTEM]
[PRESET]
[COLOR SETUP]

3 Press
[HOME]
[DISPLAY]
[DSPLY SETUP]
[3-D MAP COLORS]
[MAP COLOR]

4 Press the [NEXT] or [PREV] menu keys until you find your color system name.

5 When you display a spectrogram or color map, your color system should be represented in the data and on the color bar to the right of the display.

If your customization is not successful, an error message appears on the status line on the HP 3587S display. Make sure each parameter is entered properly in the COLORSYSTEM file.

“Undoing” a COLORSYSTEM customization

If you have simply added a group to the COLORSYSTEM file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified COLORSYSTEM file in the working directory. However, if you want to “start over again” by reverting back to the original COLORSYSTEM file in the system directory, perform the following.

1 Open a window on your operating system.

2 Remove the COLORSYSTEM file from your working directory.

3 Return to the HP 3587S window.

4 Press
[HOME]
[SYSTEM]
[PRESET]
[COLOR SETUP]

With no COLORSYSTEM file in the working directory, presetting the color setup causes the software to use the default COLORSYSTEM file in the system directory.

“Undoing” a COLORSYSTEM customization

If you have simply added a group to the COLORSYSTEM file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified COLORSYSTEM file in the working directory. However, if you want to “start over again” by reverting back to the original COLORSYSTEM file in the system directory, perform the following.

1 Open a window on your operating system.
2 Remove the COLORSYSTEM file from your working directory.
3 Return to the HP 3587S window.
4 Press
[HOME]
[SYSTEM]
[PRESET]
[COLOR SETUP]

With no COLORSYSTEM file in the working directory, presetting the color setup causes the software to use the default COLORSYSTEM file in the system directory.

8

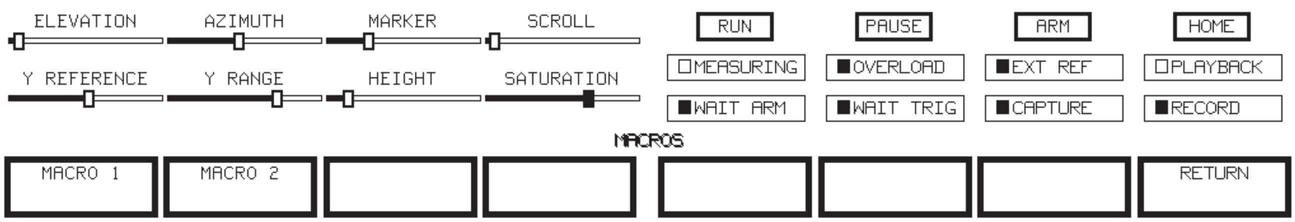
Developing a Macro

This chapter shows how to program a macro to sequentially perform several analyzer commands from one key press.

Introduction

The customization file for the macro system is called MACROSYSTEM. Macros are useful for combining several analyzer functions into one key press. For example, a macro can set the frequency span to 2 MHz, measure the peak signal and print the results to a printer when the user simply presses the softkey for that macro.

You can have up to 7 macro softkeys available at a time. The MACROSYSTEM file associates each macro with the macro softkeys which appear when you press [SYSTEM], [CUSTOMIZE] and [MACROS].



The Macro menu.

Default MACROSYSTEM* File

```
Group   : 1
Title   : MACRO 1
Command : disp_format  triple
Command : active_trace  A
Command : disp_type  spectrogram
Command : active_trace  b
Command : disp_type  spectral_map
Command : active_trace  C
Command : disp_type  spectrum
Command : grid on
Command : marker_mode  single
Command : marker_func  1
Command : active_trace  all

Group   : 2
Title   : MACRO 2
Command : disp_format  single
Command : active_trace  a
Command : disp_type  spectrum
Command : grid on
Command : marker_mode  single
Command : marker_func  1
```

* These macros are the macros initially shipped with the HP 35687B Option AGG Software. They are only included as examples and can be modified or deleted.

MACROSYSTEM Conventions

- Group titles and program commands are not case sensitive.
- Label names are displayed exactly as you enter them. You can have a two-line label. The top and bottom lines can have up to 16 characters (including spaces). A comma with no spaces separates the top line from the bottom line of a label.
- Group 1 defines the leftmost softkey in the macro menu, and groups 2 through 7 define the next consecutive softkeys to the right.
- The maximum number of groups that can be assigned is seven.
- Only commands identified in the command dictionary reference with the "MACRO" entry box checked can be used. See "Printing the Programming Reference" in chapter 2 to obtain a hardcopy listing of all commands.
- Only one command can be entered per line.
- No IF, THEN statements or other looping mechanisms are available for use in a macro.

The keywords are Group, Title, and Command, The Group keyword defines the group number and the Title defines the group title that appears as a menu selection under [SYSTEM], [CUSTOMIZATION], [MACROS]. The Command keyword identifies the commands for the macro.

Editing the MACROSYSTEM file

Before you start, make sure you complete the steps in “Copying a customization file to the work directory” in chapter 2.

1 From your <work_dir>, bring a copy of the MACROSYSTEM file into your editor.
2 If you desire, make a backup copy of these macros.
3 If you are creating a new group, open a blank line between existing groups or at the end of the last group.
4 If you are using the MACROSYSTEM template to create a new group, remove the pound sign (#) from the beginning of each line in the new group. The comments which appear after the “#” sign can be deleted or retained.
5 Enter the group number and create the macro title.
6 Enter in the desired order the commands you wish to use in the macro. When you choose macro commands, choose from those which appear in the command dictionary reference with the “MACRO” entry box checked. (“Printing the Programming Reference” in chapter 2 shows how to obtain this listing.)
7 Save the changes you just made and return to the HP-UX command line.

Activating the new MACROSYSTEM file

If MACROSYSTEM was modified while the HP 3587S was not running, starting the analyzer causes the customizations to become active in the new session. However, if MACROSYSTEM was modified during an HP 3587S session, preset the customization files by completing the following steps.

1 Return to the HP 3587S window and make sure the software is running.

2 Press
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

3 Press
[SYSTEM]
[CUSTOMIZE]
[MACROS]

4 Select the new macro label (in the default MACROSYSTEM listing, [MACRO 1]). The macro should run when you press the key.

If your customization is not successful, an error message will appear in the status line on the HP 3587S display. Make sure each command is entered exactly as shown in the command dictionary reference.

“Undoing” a MACROSYSTEM customization

If you have simply added a group to the MACROSYSTEM file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified MACROSYSTEM file in the working directory. However, if you want to “start over again” by reverting back to the original MACROSYSTEM file in the system directory, perform the following.

1 Open a window on your operating system.
2 Remove the MACROSYSTEM file from your working directory.
3 Return to the HP 3587S window.
4 Press
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

With no MACROSYSTEM file in the working directory, presetting the customization files causes the software to use the default MACROSYSTEM file in the system directory.

9

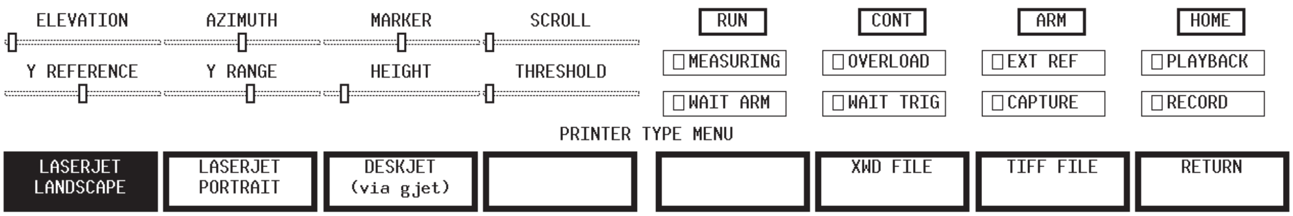
Printing to Other Types Of Printers

This chapter shows how to make the HP 35687B Option AGG software compatible with printers other than the HP LaserJet printers. It also shows how you can print data to a file.

Introduction

The customization file for printer types is called PRINTSYSTEM. This is used to make the HP 35687B work with printers other than those provided with the default system. Customizing PRINTSYSTEM works only if you provide the proper conversion program for the new printer(s) you introduce. For example, the “xpr” (X Print) program makes XWD print files compatible with the HP LaserJet, PaintJet and PaintJet XL printers. Other conversion programs are commercially available which can make XWD or TIFF print files compatible with other types of printers. The conversion program can be installed in any path listed by the $PATH command.

You can have up to 7 printer types available at a time. The PRINTSYSTEM file associates each printer with the printer type softkeys which appear when you press [HARDCOPY], [PRINTER SETUP] and [PRINTER TYPE].



The Custom Printer Type Menu

Default PRINTSYSTEM File*

```
# $Header: /stealth/s700/code/PRINTSYSTEM 1.4 1997/05/06 build Exp $
#
# Print system configuration file for HP 3587S.
#
# This file is used to configure the HP 3587S for the target print
# environment of the HP-UX workstation.
#
# For more information on printer configuration or the "gjet" utility
# (alternative for xpr), see /usr/hp3587s/README.print.
#

Group          :1
Name 1         :LASERJET
Name 2         :LANDSCAPE
Help Text      :Print screen to laser jet in landscape mode
Printer        :laser
Graphic Format :XWD
Graphic Convert:xpr -device ljet -landscape @TMPFILE
#
# NOTE: If your LaserJet printer supports the PostScript printer
#       language, you may realize better gray-scaled printouts by
#       replacing the "Graphic Convert" line above with the following:
#
#Graphic Convert:xpr -device ps -landscape -gray 3 @TMPFILE
#
Graphic Spooler:lp -or -onb -d @PRINTER
ASCII Convert  :cat @TMPFILE
ASCII Spooler  :lp -onb -d @PRINTER
ASCII Page Size:66
ASCII Format   :1

GROUP          :2
Name 1         :LASERJET
Name 2         :PORTRAIT
Help Text      :Print screen to laserjet in portrait mode
Printer        :laser
Graphic Format :XWD
Graphic Convert:xpr -device ljet -portrait @TMPFILE
Graphic Spooler:lp -or -onb -d@PRINTER
ASCII Convert  :cat @TMPFILE
ASCII Spooler  :lp -onb -d @PRINTER

GROUP          :3
Name 1         :DESKJET
Name 2         :(via gjet)
Help Text      :Print screen to deskjet in landscape mode via gjet
Printer        :deskjet
Graphic Format :XWD
Graphic Convert:/usr/hp3587s/gjet -od -mh -r -fx @TMPFILE
Graphic Spooler:lp -or -onb -d @PRINTER
ASCII Convert  :cat @TMPFILE
ASCII Spooler  :lp -onb -d @PRINTER

GROUP          :6
Name 1         :XWD FILE
Name 2         :
Help Text      :Print screen to XWD file
Printer        :
Graphic Format :XWD
```

```
Graphic Convert:mv @TMPFILE @FILENAME
Graphic Spooler:
ASCII Convert  :
ASCII Spooler  :

GROUP          :7
Name 1         :TIFF FILE
Name 2         :
Help Text      :Print screen to TIFF file
Printer        :
Graphic Format :TIFF
Graphic Convert:mv @TMPFILE @FILENAME
Graphic Spooler:
ASCII Convert  :
ASCII Spooler  :
```

* These printer types are the printer types initially shipped with the HP 35687B Option AGG Software. They are only included as examples and can be modified or deleted.

PRINTSYSTEM Conventions

- Group titles *are not* case sensitive, but printer names *are* case sensitive.
- Group titles are displayed exactly as you enter them. You can have a two-line label, specified by Name 1 and Name 2. Each line can have up to 16 characters (including spaces). This label appears in the [HARDCOPY], [PRINTER SETUP], [PRINTER TYPE] menu.
- The maximum number of groups that can be assigned is seven. Group 1 defines the leftmost softkey in the printer type menu, and groups 2 through 7 define the next consecutive softkeys to the right.
- You can enter up to 128 characters for help text.
- The Graphic Format parameter specifies whether the software should convert the graphics data to a temporary XWD file or to a temporary TIFF file. This temporary file will, in turn, be used by the Graphic Convert command.
- The Graphic Convert parameter must specify a program which converts from the specified Graphic Format (XWD or TIFF) to the graphic format of the printer. This conversion program must be installed on your system in any directory listed by the $PATH command.
- The Graphic Spooler entry pipes the converted print file to the specified destination. This destination can be a local printer or a printer which is on a LAN.
- The ASCII Convert and ASCII spooler commands are only used for printing non-graphic data to a printer such as printouts from the [PRINT HELP TEXT], [PRINT STATE] and [PRINT PRGM MANUAL] functions. Use either the cat or pprint commands to convert the ASCII data from these files to a temporary ASCII file. The ASCII Spooler entry pipes the converted ASCII file to the specified destination. This destination can be a local printer or a printer which is on a LAN.
- When you create a new group for PRINTSYSTEM, make sure you enter all the desired options for each command you use (e.g., the xpr, lp, cat and pprint commands are covered in detail in *HP-UX Reference Volume 1*.)

When the print routine runs, the graphics file is created with the converter program specified in `Graphic Convert` and stored in a temporary file. The temporary file is piped to the spooler routine specified in `Graphic Spooler`, the destination printer portion of which is specified in `Printer.`

Editing the PRINTSYSTEM file

Before you start, make sure you complete the steps in “Copying a customization file to the work directory” for the PRINTSYSTEM file in chapter 2. Also make sure you have installed the conversion program for converting XWD or TIFF files to the graphics format required by your printer. Install this program on any directory listed by the $PATH command.

1 From your <work_dir>, bring a copy of the PRINTSYSTEM file into your editor.
2 Make a backup copy of this default group configuration.
3 If you are creating a new group, open a blank line between the end of the first group and the new group.
4 Enter the group number and softkey label for the printer type menu.
5 Enter the help text for printer type selection you are creating (up to 128 characters).
6 If you creating a group for a different type of printer, enter the following:

- Printer name
- Graphic Format required for that printer (either XWD or TIFF)
- The conversion command for converting the display graphics data to the format of the specified printer. (Make sure you specify the desired options for the conversion).
- The print destination for the graphic spooler.
- The conversion command for converting the ASCII data to the specified printer. (Make sure you specify the desired options for the conversion).
- The print destination for the ASCII spooler.

7 Save the changes you just made and return to the HP-UX command line.

If you want to print to non-HP color printers, the following table describes the type of monitor and refresh rate needed for this kind of printing.

Activating the new PRINTSYSTEM file

If PRINTSYSTEM was modified while the HP 3587S was not running, starting the analyzer causes the customizations to become active in the new session. However, if PRINTSYSTEM was modified during an HP 3587S session, preset the customization files by completing the following steps.

1 Return to the HP 3587S window and make sure the software is running.

2 Press
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

3 Press
[HARDCOPY]
[PRINTER SETUP]
[PRINTER TYPE]

4 Select the new printer (in the example, [DESKJET]). You should now be able to print to the new printer.

If your customization is not successful, an error message will appear in the status line on the HP 3587S display. Make sure you have provided all the parameters needed for the new printer and that the conversion program is installed in the proper directory.

"Undoing" a PRINTSYSTEM customization

If you have simply added a group to the PRINTSYSTEM file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified PRINTSYSTEM file in the working directory. However, if you want to "start over again" by reverting back to the original PRINTSYSTEM file in the system directory, perform the following.

1 Open a window on your operating system.
2 Remove the PRINTSYSTEM file from your working directory.
3 Return to the HP 3587S window.
4 Press
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

With no PRINTSYSTEM file in the working directory, presetting the customization files causes the software to use the default PRINTSYSTEM file in the system directory.

Printing data to a file

If you want to include a printout of results in a report using a desktop publisher, print the data to a file. The default PRINTSYSTEM file provides for printing to X Windows (XWD) files and TIFF files. These are common file formats for most desktop publishers. If you desire to print to a different file format, follow the first three steps in the following exercise which describe how to create a new group that prints data to a different kind of file than XWD or TIFF. The remaining steps show how to use the HP 35687B to print data to a file.

1 Skip to step 4 if you are printing to an XWD or TIFF file. If you are creating a new group that will print data to a file other than an XWD or TIFF file, follow steps 1 through 5 in “Adding a printer to the PRINTSYSTEM file”. Then enter the following two parameters to the group — the remaining parameters can be left blank.
a. Enter the desired Graphics Format for the file (i.e., XWD, TIFF)
b. For Graphics Convert, enter the conversion command for converting the display graphics data (XWD) to the desired file format. (Make sure you specify the desired options for the conversion).

2 Save the changes you just made and return to the HP-UX command line.

3 Activate the PRINTSYSTEM file. See “Activating the new PRINTSYSTEM file”.

4 Press
[HARDCOPY]
[PRINTER SETUP]
[PRINTER TYPE]
Press [XWD FILE], [TIFF FILE] or, if you created a new file type, press the softkey with the new name.
[RETURN]

5 Press
[FILENAME]
Type the desired file name and press `<Return>`

6 Press
[HARD DISK UTILITIES]
[FILE TYPE]
Press [XWD FILE], [TIFF FILE] or, if you created a new file type, press the softkey with the new name.

7 Press
[RETURN]
Toggle [WRT PRTCT] to [OFF]
[RETURN] .

8 Set up the display the way you want it and press [PRINT SCREEN]. See “Printing data to a printer” in the *HP 35687B Operator's Reference* for more information on printing.

10

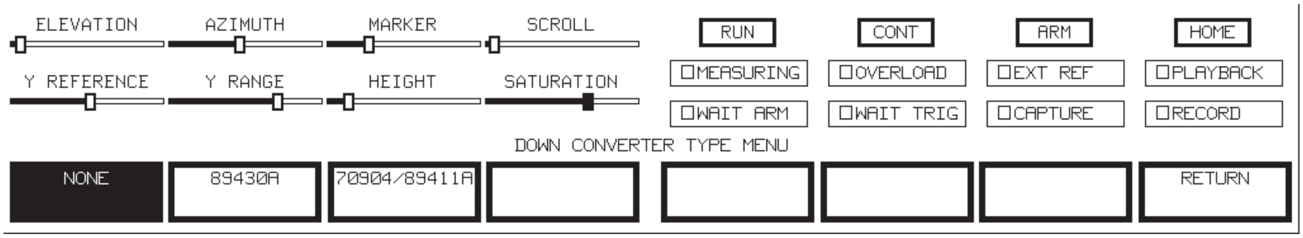
Using a Downconverter

This chapter shows you how to set up the menu system to support using downconverter.

Introduction

The customization file for downconverter types is called DOWNCONVERTER. This is used to configure the downconverter to work with the HP 35687B.

You can have as many as 7 downconverter types defined and active at a time. The DOWNCONVERTER file associates each downconverter with the softkeys which appear when you press [MEASUREMENT], [FREQUENCY SETUP], [DOWNCNVRTR SETUP], [DOWN CONVRTR].



The Custom Downconverter Type Menu.

Default DOWNCONVERTER File*

```
GROUP           :1
Name1           :NONE
Name2           :
Help Text       :No down converter is selected to be controlled
Help Text       :from the 3587s user interface.
Interface       :None
Timeout         :0
LO Offset       :0.0
IF Output CF    :0.0
Input CF        :0.0
Attenuation     :0.0
Mirror          :0
Gain            :0


GROUP           :2
Name1           :89430A
Name2           :
Help Text       :The 89430A is a 3 MHz to 1.8 GHz down converter with a
Help Text       :nominal 17 dB of gain.  The input center frequency
Help Text       :has a 1 MHz resolution.  The attenuation settings
Help Text       :are in 5 dB steps up to a total of 55 dB.
Interface       :e1485_rs232
Device File     :
```

```
baud rate       :9600
Init Command    :)Y\n)Q\nt5\n
Atten Command   :a%.0f\n
LO Command      :f%.0f\n
LO Offset       :-3000000.0
LO Output Scalar:1.0e-6
IF Output CF    :3000000.0
Input CF        :100000000.0
Input CF Step   :1000000.0
Min Input CF    :3000000.0
Max Input CF    :1793000000.0
Attenuation     :0.0
Atten Min       :0
Atten Max       :55
Atten Step      :5
Mirror          :1
A/D Range       :16
Gain            :17


GROUP           :3
Name1           :70910/89411A
Name2           :
Help Text       :The 70910/89411A is a 1 MHz to 26.5 GHz down converter
Help Text       :with a nominal 10 dB of gain.  The input center frequency
Help Text       :has a 1 Hz resolution.  The attenuation settings
Help Text       :are in 10 dB steps up to a total of 70 dB.
Interface       :HPIB
Address         :18
Timeout         :5
Init Command    :FOFFSET -3.6214 GZ; SS; LN;
Atten Command   :PTOPEN 6,18;PT /ATGN -%.0f;/;PTCLOSE;
LO Command      :CF %16.0f; SP 0 HZ; LN; TS;
LO Offset       :2600000.0
IF Output CF    :3000000.0
Input CF        :100000000.0
Input CF Step   :1.0
Min Input CF    :1000000.0
Max Input CF    :26500000000.0
Attenuation     :0.0
Atten Min       :0
Atten Max       :70
Atten Step      :10
Mirror          :0
A/D Range       :16
Gain            :10           # 89411A set to +5 dB


GROUP           :4
Name1           :89431A
Name2           :
Help Text       :The 89431A is a 3 MHz to 2.65 GHz down converter with a
Help Text       :nominal 17 dB of gain.  The input center frequency
Help Text       :has a 1.171875 MHz resolution.  The attenuation settings
Help Text       :are in 5 dB steps from -20 dB to +60 dB.
Interface       :e1485_rs232
Device File     :
baud rate       :9600
Init Command    :)Y\n)Q\nt5\n
Atten Command   :a%.0f\n
LO Command      :fn%.0f\n
LO Offset       :3042.484375e6
LO Output Scalar:.8533333333e-6
IF Output CF    :2.484375e6
```

```
Input CF        :100.484375e6
Input CF Step   :1.171875e6
Min Input CF    :3.0e6
Max Input CF    :2.653e9
Attenuation     :0.0
Atten Min       :-20
Atten Max       :60
Atten Step      :5
Mirror          :1
A/D Range       :16
Gain            :17


GROUP           :5
Name1           :WJ9119
Name2           :
Help Text       :The WJ9119 is a 500 kHz to 32 MHz downconverter with a
Help Text       :nominal 9 dB of gain.  The input center frequency
Help Text       :has a .25 MHz resolution.  The attenuation settings
Help Text       :are in 1 dB steps up to a total of 47 dB.
Interface       :software
Address         :140142
Device File     :
Baud Rate       :
Init Command    :
Atten Command   :
LO Command      :
LO Offset       :0.0
LO Output Scalar:1.0
IF Output CF    :2.56e6
Input CF        :10e6
Input CF Step   :250e3
Min Input CF    :.25e6
Max Input CF    :32e6
Attenuation     :6
Atten Min       :0
Atten Max       :47
Atten Step      :1
Mirror          :1
A/D Range       :-8
Gain            :21
```

* These types are initially configured in the HP 35687B Option AGG Software. They are only included as examples and can be modified or deleted.

DOWNCONVERTER Conventions

- Downconverter names *are* case sensitive.
- You can have a two-line label, specified by `Name1` and `Name2`. Each line can have up to 16 characters (including spaces). This label appears in the [MEASUREMENT], [FREQUENCY SETUP], [DOWNCNVRTR SETUP], [DOWN CONVRTR] menu.
- The maximum number of groups that can be assigned is seven. Group 1 defines the leftmost softkey in the printer type menu, and groups 2 through 7 define the next consecutive softkeys to the right.
- You can enter up to 256 characters for help text.
- `Interface` specifies the interface type used to connect the computer to the analyzer/downconverter. Allowable entries are HPIB, RS232, E1485_RS232, Software and None. Others may be added to this list in the future.
- `Address` specifies the downconverter's HP-IB interface address used to control the downconverter. (This parameter is also passed when opening the software interface.)
- `Device File` specifies the device file used to communicate with a particular interface. The default device file for the RS-232 interface is /dev/rs232.
- `Baud Rate` specifies the rate to transmit data over the RS-232 interface. The supported baud rates are (in bps) 50, 75, 110, 134, 150, 200, 300, 600, 900, 1200, 1800, 2400, 3600, 4800, 7200, 9600, 19200, and 38400. (This does not affect the E1485_RS232 interface.)
- `Timeout` specifies the delay (in seconds) that the controller waits for the analyzer/downconverter to accept commands before the controller aborts the command and reports a timeout error.
- `Init Command` specifies the initialization command string used to initialize the analyzer/downconverter when it is first selected. If the null string is entered, no initialization commands are sent. The Init, Attn, and LO Commands are all sent when the downconverter is selected.
- `Attn Command` specifies the attenuation command used to initialize the attenuator, including the format used to specify the attenuation value. If the null string is entered, no attenuation commands are sent. The Init, Attn, and LO Commands are all sent when the downconverter is selected.
- `LO Command` specifies the local oscillator command and numeric format used to initialize the downconverter's local oscillator. If the null string is entered, no LO commands are sent. The Init, Attn, and LO Commands are all sent when the downconverter is selected.
- `LO Output Scalar` specifies a scalar that the LO should be multiplied by before being sent via the specified interface to the downconverter; it allows you to send a number representing frequency units such as MHz or kHz without having to send a string of zeros. The equation for the LO frequency value (sent in `LO Command`) is as follows:

$$LO\_frequency = (Input\_CF + LO\_offset) \times LO\_output\_scalar$$

- where LO_offset and LO_output_scalar are in the DOWNCONVERTER file, and Input_CF is a value entered from the Downconverter Setup Menu.
- Example: The HP 89430A accepts the LO in units of MHz. The LO offset is set to –3 MHz and the LO output Scalar is set to $1.0E^{-6}$ for a 100 MHz input CF, the LOfreq sent to the HP 89430A would be 97.

- `LO Offset` specifies the local oscillator frequency offset (in Hz). It is the LO frequency when the input center frequency is tuned to 0 Hz (when the IF-Out CF is correctly set to the downconverter's specified output center frequency).
- `IF Output CF` specifies the frequency within the 4 MHz bandwidth of the E1430 (ADC VXI module) that corresponds to the Input `CF.` This allows the E1430 center frequency to be set to a value other than the `IF Output CF`.
- `Input CF` specifies the down converter's input center frequency (in Hz).
- `Min Input CF` specifies the lowest possible input center frequency (in Hz).
- `Max Input CF` specifies the highest possible input center frequency (in Hz).
- `Attenuation` specifies the initial attenuation setting (in dB).
- `Atten Min` specifies the lowest possible attenuation setting (in dB).
- `Atten Max` specifies the highest possible attenuation setting (in dB).
- `Atten Step` specifies the attenuation step size (in dB).
- `Mirror` specifies whether to invert the spectrum (low frequencies moved to high end and high frequencies moved to low end). Entries are either 0 (mirror function inactive) or 1 (mirror function active).
- `ADC Range specifies` the analog-to-digital converter range in units of dBm, that the ADC is set to when this downconverter type is selected.
- `Gain` specifies the gain of the downconverter, in dB, when the attenuator setting is 0 dB. The combination of this gain value and the current attenuator setting is used to compute an appropriate amplitude scalar as shown in the following equation:

$$amplitude\ scalar = 10\left(\frac{gain - attenuation}{-20.0}\right)$$

- When you create a new group for DOWNCONVERTER, make sure you enter all the desired options for each parameter you use.

Command Strings

In the Init, Atten, and LO Command strings, special characters are used. The value calculated for the attenuation and LO values are double-precision (64-bit), floating-point numbers. These numbers are added to the command strings by using the print f(3) formatting strings. See Section 3 of the HP-UX Reference, Volume 2. A common example is of the form "%.0f" which specifies a number format with no digits to the right of the decimal point.

Editing the DOWNCONVERTER file

Before you start, make sure you complete the steps in “Copying a customization file to the work directory” for the DOWNCONVERTER file in chapter 2.

1 From your <`work_dir`>, bring a copy of the DOWNCONVERTER file into your editor.
2 Make a backup copy of this default group configuration.
3 Enter the group number and create the user title. Specifying group number 1 makes that group part of the system default. Group numbers 2 through 7 must be activated by the user after the HP 35687B software is running. (See “Activating a customization”.)
4 Enter settings for the various fields as described in the following:
5 Save the changes you just made and return to the HP-UX command line.

Activating the new DOWNCONVERTER file

If DOWNCONVERTER was modified while the HP 3587S was not running, starting the analyzer causes the customizations to become active in the new session. However, if DOWNCONVERTER was modified during an HP 3587S session, preset the customization files by completing the following steps.

1 Return to the HP 3587S window and make sure the software is running.

2 Press
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

3 Press
[SYSTEM]
[CUSTOMIZE]
[KEYBOARD]

4 Select the new menu system label (in the example, [ACCEL SYS 2]). The keyboard accelerator system should now operate using your customized file.

If your customization is not successful, an error message will appear in the status line on the HP 3587S display. Make sure each command is entered exactly as shown in the command dictionary reference.

“Undoing” a DOWNCONVERTER customization

If you have simply added a group to the DOWNCONVERTER file or slightly modified one of the existing groups, you can delete or comment out the undesired changes by editing the modified DOWNCONVERTER file in the working directory. However, if you want to “start over again” by reverting back to the original DOWNCONVERTER file in the system directory, perform the following.

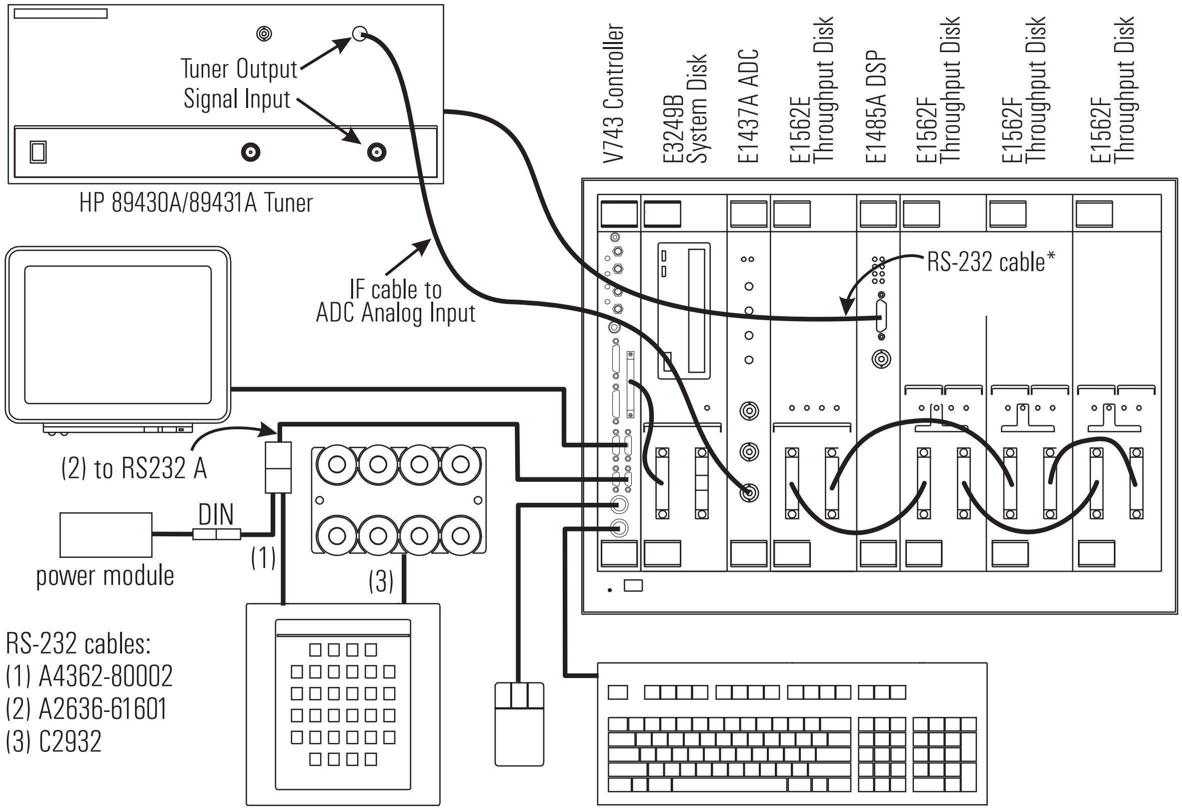
1 Open a window on your operating system.

2 Remove the DOWNCONVERTER file from your working directory.

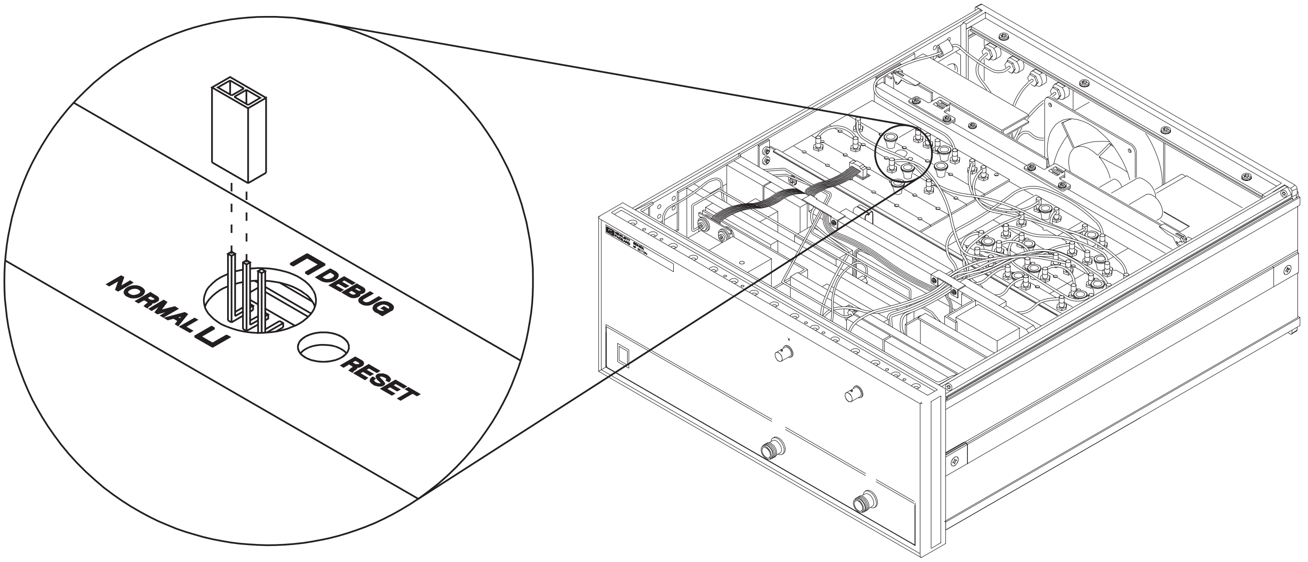
3 Return to the HP 3587S window.

4 Press
[SYSTEM]
[PRESET]
[CUSTOMIZE FILES]

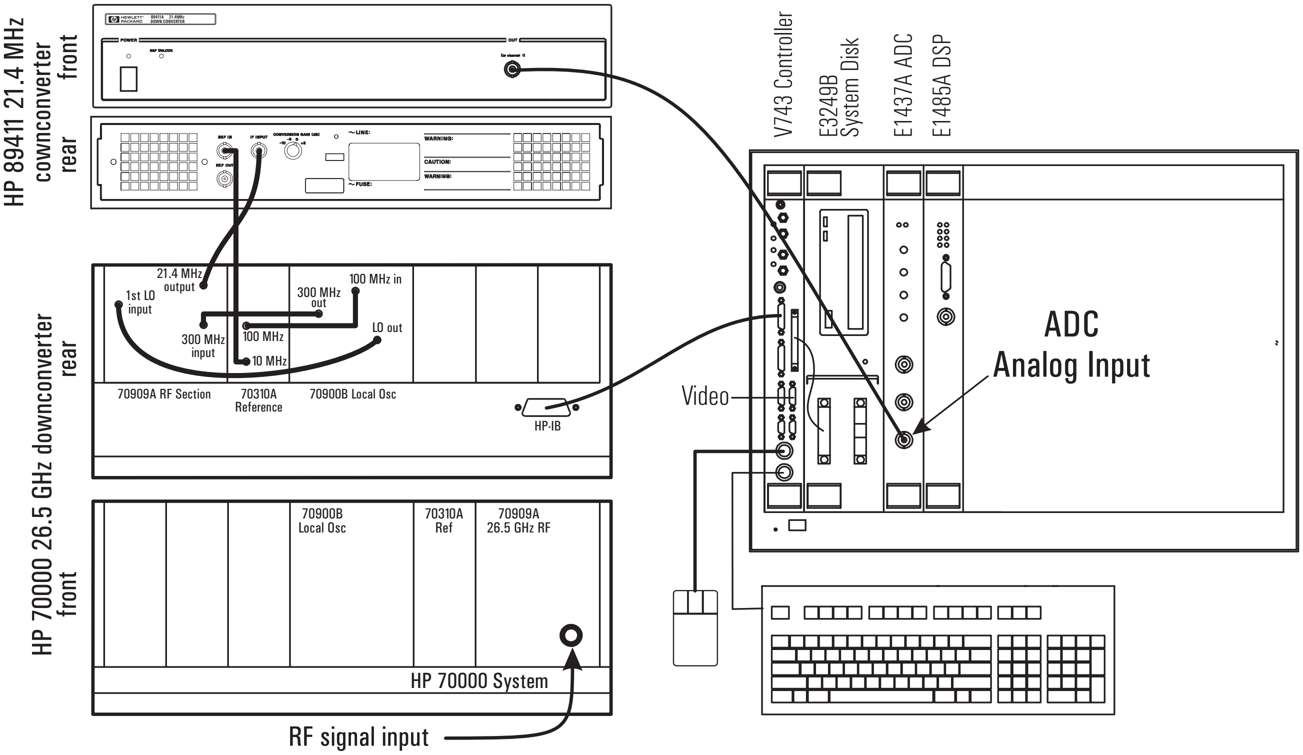
With no DOWNCONVERTER file in the working directory, presetting the customization files causes the software to use the default DOWNCONVERTER file in the system directory.



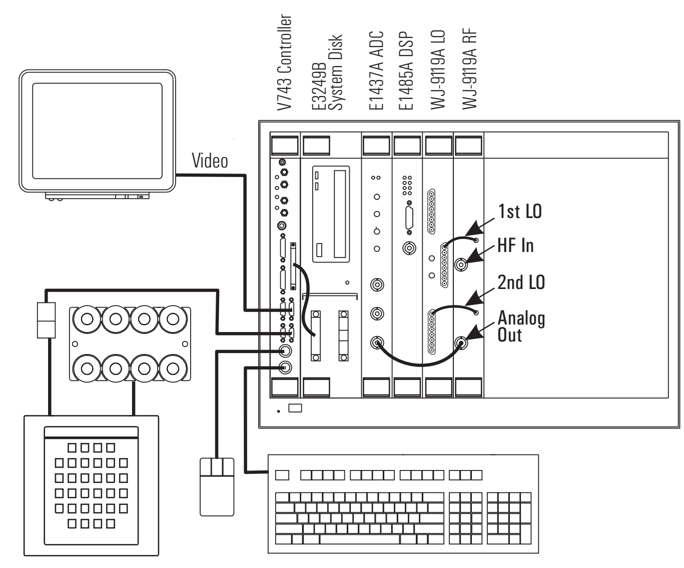
Adding the HP 89430A /89431A downconverter to the system.
***See Hardware Installation Note concerning the RS232 control cable.**



Move jumper to the DEBUG position before using the downconverter.



Configuration of the 26.5 GHz downconverter.



For HF (2 MHz to 32 MHz) use the Watkins-Johnson WJ-9119.

11

Command Port

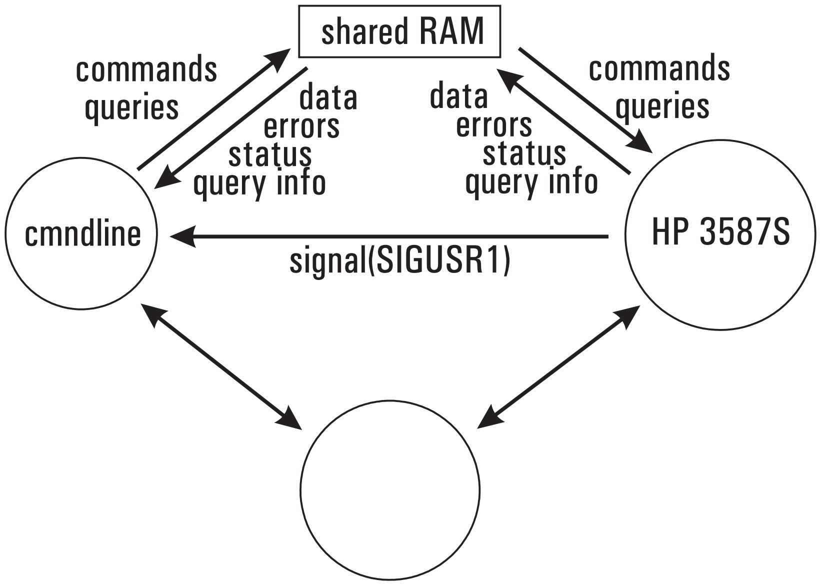
This chapter describes how to control the HP 35687B software with another program. This is known as the command port. This discussion covers use of the program cmndLine.c which is included in the system software.

Introduction

The command port allows the user to access to the power and flexibility of the HP3587s Realtime Signal Analyzer from another HP-UX program. You can send the same commands used in customizing the knob box, button box, and other user input features to control the analyzer, and you can also read data back into your program for further unique processing.

The architecture of the command port in shown in the figure below. The mechanism for communication between the HP3587s process and your process is shared RAM. Shared RAM is a dedicated area of memory that can be accessed by two or more processes. For further information consult your HP-UX manuals and also look specifically at the commands shmctl(2), shmget(2), shmat(2), and shmdt(2).

This shared RAM segment is configured to accept commands and queries from your process and return data, errors, status information, and/or query responses.



Process Diagram for Automated Control with cmndLine

External Control Commands

There are seven fundamental commands associated with the command port.

int openCommandPort(int key)

This command opens communication with the command port. If the command is successful a 0 is returned, else a -1 is returned. The “key” parameter specifies which HP 3587S process to communicate with via HP-UX shared RAM. (See “Command Port Key” chapter 1.

void closeCommandPort(void)

This command ends communication with the command port.

int sendCommand(char *command)

This command writes a string to the command port and returns immediately. If successful, a 0 is returned. If a -1 is returned, either the command port wasn't opened or the command port already has a command in its command buffer.

int waitForCommandComplete(double timeout, double interval)

This command waits 'timeout' seconds for the previous command to be processed by the HP3587s system. It checks every 'interval' seconds. The 'interval' time must be less than a second in duration. This command allows you to synchronize the command stream with the HP3587s process. If the command is completed before 'timeout' seconds a 0 is returned, else a -1 is returned.

void readErrorString(char *string)

This command returns the current error string. The error string consists of a comma delimited series of error codes that have occurred since the last time the error string was read. The maximum length of this string is ERROR_LENGTH. The error codes are described later in this chapter. If no errors are present, a zero length string is returned. After each call, the error buffer is cleared.

void readQuery(char *string)

This command returns the current query response. The maximum length of any query is QUERY_LENGTH. The query buffer is truncated after each call of this function. If a query command is sent, and the response is not read, the next query command overwrites the previous query command.

```
void getDataPtrs(void **header, void **status, void **timeStamp,
                 void **spectrum, void **histogram, void **histogramCount,
                 void **realTime, void **imagTime)
```

This command returns the pointers to various data results passed back to your process. If any of the passed values are NULL, no pointer for that value is returned. If the command port has not been opened or if the data is invalid, a -1 is returned. If the data is valid a 0 is returned. Each requested pointer should be checked to be sure it is not NULL before using it. If the data for a particular data type is requested, and the data is not available, a NULL pointer is returned.

Examples

To Send a Command

To send the command ‘run’ to the hp3587s process, the following pseudo-code could be used:

```
main()
{
   openCommandPort(KEY);
   sendCommand("run");
   waitForCommandComplete(5.0, 0.1);
   closeCommandPort();
}
```

Note that error checking has not been shown for simplicity and brevity.

To Query a Value

To query the current elevation of trace C from the hp3587s process, the following pseudo-code could be used:

```
main()
{
   char queryResonse[QUERY_LENGTH];

   openCommandPort(KEY);
   sendCommand("elevation?");
   waitForCommandComplete(5.0, 0.1);
   readQuery(queryResponse);
   printf("Elevation Query: %s\n",queryResponse);
   closeCommandPort();
}
```

The return string will be 5, 5, 5 which is the current value of the elevation for trace A, B, and C, respectively.

Reading Errors

To read error codes from mis-typed commands, the following pseudo-code could be used:

```
main()
{
   char errorResonse[ERROR_LENGTH];

   openCommandPort(KEY);
   sendCommand("actve_trace C");   correct spelling
   waitForCommandComplete(5.0, 0.1);
   readError(errorResponse);
   printf("Error: %s\n",errorResponse);
   closeCommandPort();
}
```

The error returned is 36. See error messages at end of this chapter.

Reading Data

The following pseudo-code demonstrates how to read spectrum data from the HP3587s process via the command port.

```
main()
{
   struct aDataHeader *dh;
   float *spectrum;
   int i;

   openCommandPort(KEY);
   sendCommand("read_data 0");
   waitForCommandComplete(5.0, 0.1);
   getDataPtrs((void **)&dh,0,0,(void **)&spectrum,0,0,0,0);
   if(spectrum){
      for(i=0;i<dh->numLines;i++){
         printf("Point:%d Value:%f\n",i,*spectrum++);
      }
   }
   closeCommandPort();
}
```

If you want to continually read and print the data values, you could put a while loop in the previous program.

```
 main()
 {
    struct aDataHeader *dh;
    float *spectrum;
    int i;

    openCommandPort(KEY);
    while(1){
       sendCommand("read_data 0");
       waitForCommandComplete(5.0, 0.1);
       getDataPtrs((void **)&dh,0,0,(void
**)&spectrum,0,0,0,0);
       if(spectrum){
          for(i=0;ih-numLines;i++){
             printf("Point:%d Value:%f\n",i,*spectrum++);
          }
       }
    }
    closeCommandPort();
 }
```

This program has one major problem: if you are reading data faster than it is being processed by the HP3587s process, you may reread the same data block. You could always do some type of checking, but doing so degrades performance. To solve this problem, we can use another feature of the command port: signals. When the command 'data_update' is sent with the process id of your process, the hp3587s process sends a SIGUSR1 signal to your process every time new data is available.

The following pseudo-code shows a simple example:

```
void dataEvent(void);
int dataPending = 0;
struct sigvec dataVec = {dataEvent,0,0};

main()
{
   struct aDataHeader *dh;
   float *spectrum;
   int i, pid;
   char string[80];

   pid = getpid();
   openCommandPort(KEY);
   sigvector(SIGUSR1,&dataVec,(struct sigvec *)0);
   sprintf(string,"data_update %d",pid);
   sendCommand(string);
   waitForCommandComplete(5.0, 0.1);

   while(1){
      usleep(5000);
      if(dataPending){
         dataPending = 0;
         sendCommand("read_data 0");
         waitForCommandComplete(5.0, 0.1);
         getDataPtrs((void **)&dh,0,0,(void **)&spectrum,0,0,0,0);
         if(spectrum){
            for(i=0;ih-numLines;i++){
               printf("Point:%d Value:%f\n",i,*spectrum++);
            }
         }
      }
   }
   closeCommandPort();
}

void dataEvent(void)
{
   dataPending = 1;
}
```

For more documentation on signals see you HP-UX reference manuals and specifically look at sigvector(2) and signal(5) manual pages. If you want to turn off the signal from the HP3587s process, send the command 'data_update 0'.

By default, the HP3587s process sends the SIGUSR1 signal, however this can be changed to SIGUSR2 signal by sending the command 'pp_signal'.

Also by default, the HP3587s process sends the signal asynchronously, meaning that once the signal is sent, the HP3587s process continues to gather and process new data. If you want to synchronize the two processes sent the command 'pp_xfermode sync'. The HP3587s process waits for your process to send the 'read_data' command before continuing. The command 'pp_timeout' specifies the time the HP3587s process waits before gathering the next block of data in the synchronous transfer mode.

Running the cmndLine Program

Included on the disk in the /usr/hp3587s/src/commandPort/ directory is an example program called cmndLine. This program allows you to type in the commands found in the programming reference manual and display error and query results. It also displays spectrum data. To run this program:

1. Be sure that the analyzer is set up to turn on with the HP-UX login rather than running immediately. See chapter 1 for details on how to do this.
2. login as hp3587s <Return>
3. Start the X11 windows user interface. This is not required if the workstation is running HP VUE or CDE (common desktop environment).
   x11start <Return>
4. hp3587s -t0 & <Return>
   This runs the analyzer as a small window in a background process.
5. cd /usr/hp3587s/src/commandPort<cmndLine>
6. cmndLine <Return>
7. Note the pid number printed on the screen.
8. At the prompt, type: disp_format dual <Return>
9. To display data, type read_data 0 <Return>
   An X window displaying a trace appears; you should move it so it doesn't overlay the HP3587s window.
10. To continually update via signals type: data_update <pid> <Return> where <pid> is the value noted in the a previous step.
11. To turn on the signals, type data_update 0 <Return>
12. To exit, type quit <Return>

To Modify the cmndLine program

If you wish to modify the cmndLine program, you should copy the entire /usr/hp3587s/src/commandPort directory to your user area such as $HOME/commandPort. Perform the following steps to do this:

1. cd <Return>
2. mkdir commandPort <Return>
3. cd commandPort <Return>
4. cp /usr/hp3587s/src/commandPort/* . <Return>
   to recompile the code, type: make <Return>

Error messages

Error numbers may be translated into messages with the following table.

Error Messages

Error Number	Message
1	Couldn't open file.
2	Invalid descriptor.
3	No descriptor found.
4	Group number out of range.
5	Unknown printer format.
6	No Group number assigned.
7	Unsupported down converter interface.
8	Too many errors.
9	Color mode is invalid.
10	Hue is valid for COLOR mode only.
11	No Hue value was found.
12	Intensity is valid for GRAYSCALE mode only.
13	No Intensity value was found.
14	Number of Colors is valid for USER mode only.
15	No 'Number of Colors' value was found.
16	No color index has been specified.
17	No 'Number of Colors' has been specified.
18	Color number is out of range.
19	3 color values are required.
20	Macro group has already been used.
21	Macro memory is full.
22	No Button number specified.
23	Button number is out of range.
24	No knob number specified.
25	Knob number is out of range.
26	No menu number specified.
27	Menu number is out of range.
28	No key number specified.
29	Key number is out of range.
30	No menu value has been specified yet.
31	Incorrect number of parameters.
32	Internal compiler error.
33	Command not available.
34	Parameter out of range.
35	Unrecognized parameter.
36	Unrecognized command.
37	Label field is too long.
38	Command is query-only.
39	Command cannot be queried.
1000	No valid data for read_data request
1001	Invalid index for read_data request
1002	Not enough RAM for read_data request

12

Signal Capture Data Headers

This chapter defines the header data used in signal capture files. This format differs from the header format used for standard data files. (see next chapter).

Signal Capture Data Structure

A signal capture file consists of this header with 16-bit or 32-bit integer data appended to the end.

The following code appears in sigCap.h:

```
struct aSignalCaptureHeader{
   char   version[8];
   int    valid;
   int    type;
   double startFreq;
   double stopFreq;
   double centerFreq;
   double span;
   double freqScalar;
   double freqMultiplier;
   int    refFreqMode;
   double extRefFreq;
   double fs;
   int    zoom;
   int    filterPass;
   double cfRes;
   double cfBits;
   int    bandwidth;
   int    mirror;
   int    coupling;
   int    analogFilterBypass;
   int    inputSource;
   int    reserved;
   int    dataPrecision;
   int    inputRangeIndex;
   int    signalCaptureSize;
   double trigLevel;
   double trigMagLevel;
   int    trigSlope;
   int    trigMode;
   double trigDelay;
   double userScalar;
   char   parm_pad[152];
   int    overload;
   int    numSamples;
   int    numDataBytes;
   double startTime;
   char   pad[156];
};
```

Parameter Definitions

The following is a description of each parameter field:

version[8]	This field contains the current version string of the signal capture header. The current version is A.02.00. This version changes if the header is no longer compatible with previous definitions of the header. The software will always be able to process previous versions of the header, but not necessarily further version.
valid	If this field is set, the associated data is valid. If it is zero, the data is invalid.
type	Currently not used. Always set to zero.
startFreq	The start frequency setting, in Hz, of the ADC when this data was captured.
stopFreq	The stop frequency setting, in Hz, of the ADC when this data was captured.
centerFreq	The center frequency setting, in Hz, of the ADC when this data was captured.
span	The span frequency setting, in Hz, of the ADC when this data was captured.
freqScalar	The frequency scalar frequency setting, in Hz, of the ADC when this data was captured. The startFreq, stopFreq, centerFreq, and centerFreq all include this value.
freqMultiplier	The frequency multiplier value active when this data was captured. The startFreq, stopFreq, centerFreq, and centerFreq all include this value.
refFreqMode	The reference frequency mode of the ADC. The possible values are: SC_REF_FREQ_INTERNAL SC_REF_FREQ_EXTERNAL
extRefFreq	External sample clock frequency in Hz.
fs	The external reference (sample) frequency, in Hz, of the ADC.
zoom	Specifies whether the ADC mixed the input ADC samples with a digital local oscillator. If they were mixed, the data stored in the file will be complex. If they were not mixed, the data is baseband data. The possible values are: SC_BASEBAND SC_ZOOM
filterPass	The number of digital filter passes used to process the data.
cfRes	The frequency resolution capability of the center frequency setting. This is calculated with: `cfRes = sample clock/SC_CF_BIT_FACTOR` If the sample clock is 10.24 MHz, the center frequency resolution is 0.000010 Hz.
cfBits	The number of center frequency resolution steps. This field is used to specify the center frequency value. The center frequency is calculated with the following equation: `center frequency = cfBits * cfRes`

bandwidth	Specifies whether the data is digitally filtered to fs/2 (full bandwidth) or fs/4 (half bandwidth), where fs is the effective sample clock rate. The possible choices are: SC_FULL_BANDWIDTH SC_HALF_BANDWIDTH
mirror	Specifies whether the input signal is the lower half of a mixed image or the upper half. When the signal is the lower half, the spectrum appears on the X axis with the highest frequency at the left and the lowest frequency at the right. Turning mirroring on reverses this, making the spectrum normal (lowest frequencies at left). The possible choices are: SC_MIRROR_OFF SC_MIRROR_ON
coupling	Specifies whether the ADC was AC-coupled or DC-coupled when the data was captured. The possible choices are: SC_DC_COUPLED SC_DC_COUPLED
analogFilterBypass	Specifies whether the ADC analog anti-alias filter was bypassed when the data was captured. The possible choices are: SC_ANALOG_BYPASS_OFF SC_ANALOG_BYPASS_ON
inputSource	Specifies whether the ADC used single-ended or differential grounding when the data was captured. The possible choices are: SC_INPUT_SOURCE_SINGLE SC_INPUT_SOURCE_DIFF
dataPrecision	Specifies whether the data is stored as 16-bit or 32-bit data. The possible choices are: SC_DATA_PRECISION_16 SC_DATA_PRECISION_32
inputRangeIndex	Specifies the input range of the ADC. A zero is the maximum range value (28 dBm), ten is the lowest range value (-32 dBm), and each step is 6 dBm..
signalCaptureSize	Specifies the size of the signal capture buffer (in samples) when the data was captured.
trigLevel	Specifies the trigger level in % of full scale.
trigMagLevel	Specifies the magnitude trigger level in dBm below full scale.
trigSlope	Specifies the trigger slope. The possible choices are: SC_TRIG_SLOPE_POS SC_TRIG_SLOPE_NEG
trigMode	Specifies the trigger mode used when the data was captured. The possible choices are: SC_TRIG_MODE_FREE SC_TRIG_MODE_LEVEL SC_TRIG_MODE_MAGNITUDE SC_TRIG_MODE_EXT
trigDelay	Specifies the trigger delay in number of sample.
userScalar	The user amplitude scalar specified when the data was captured. This value is not included in the data values.

overload	Specifies whether an overload of the ADC occurred during capture of this data. The possible values are: SC_NO_OVERLOAD SC_OVERLOAD
numSamples	Specifies the number of data samples stored with this header.
numDataBytes	Specifies the amount of data, in bytes, stored with this header.
startTime	Specifies the capture start time, in seconds, since the last time the RUN key was pressed.

Scaling Data

To scale the integer data stored in the signal capture file, the following code segment should be used:

```
scalar = inputRange[inputRangeIndex] / 0.433430;
if(zoom) scalar *= 2;
scalar *= userScalar;
if(dataPrecision == SC_DATA_PRECISION_32)
   scalar /= 2147483648.0;
else
   scalar /= 32768.0;
```

where `inputRange[]` is defined as:

```
double inputRange[] = {8.0, 4.0, 2.0, 1.0, 0.5, 0.25, 0.125,
                      0.0625, -0.03125, -0.015625, -0.0078125};
```

Converting Data Formats

Two utilities are provided for converting HP3587S signal capture files to and from the Standard Data Format (SDF) file format. The SDF format is used for transferring data between many of the HP newer signal analyzers. Some of those that support the Signal Capture file from the HP3587S are:

HP89410
HP89440
HP35655A
HP35670A
HP3566A
HP3567A

Once your files are in the SDF format, utility programs are available to convert SDF to other common formats. These formats are:

- PC-MATLAB, a trademark of The MathWorks, Inc., is a software package for general digital signal processing and filtering.
- MATRIXx, a product of Integrated Systems Inc., is a software package for control system analysis.
- ASCII is a versatile format for spreadsheets and other general software.

For the complete documentation of SDF and a copy of the SDF Utilities, order the *Standard Data Format Utilities: User Guide*.

The two utility programs provided with the HP3587S to convert between data types are SCtoSDF and SDFtoSC.

SCtoSDF

The SCtoSDF utility is used to convert from a Signal Capture file to a SDF file. To convert the file capture.cap to capture.tim, type the following:

```
SCtoSDF -fcapture
```

SDFtoSC

The SDFtoSC utility is used to convert from a SDF file to a Signal Capture file. To convert the file capture.tim to capture.cap, type the following:

```
SDFtoSC -fcapture
```

For information about additional options for each of these utilities, use the -u command line option to display the usage.

13

Data Headers

This chapter defines the header data used to define data structures in marker functions, display functions, and data stored on the disk when you save a file or record a measurement. The data header format differs from the header format used for signal capture data (see previous chapter).

Header Data Structure

The header consists of two sections.

The first is the general information describing the active settings, such as span and input range , when the data was stored.

The second is the offsets of each particular data type in the data section.

The following code appears in header.h:

```
struct aDataHeader{
   char   version[8];
   int    valid;
   int    changeFlag;
   /*** FREQUENCY PARAMETERS ***/
   int    numLines;
   double startFreq;
   double deltaF;
   /*** TIME PARAMETERS ***/
   int    blocksize;
   double startTime;
   double deltaT;
   /*** HISTOGRAM  PARAMETERS ***/
   int    histBlkSize;
   double startVoltage;
   double deltaV;
   /*** PAD **********************/
   char   parmPad[128];
   /*** MEASUREMENT PARAMETERS ***/
   double span;
   double centerFreq;
   double timeLength;
   int    zoom;
   double inputVoltage;
   int    inputRange;
   int    numAverages;
   int    overlap;
   int    measResults;
   double fs;
   double trigLevel;
   double trigMagLevel;
   double trigDelay;
   int    bandwidth;
   int    coupling;
   int    analogFilterBypass;
   int    inputSource;
   int    dataPrecision;
   int    trigSlope;
```

```
   int    trigMode;
   int    windowType;
   int    averageMode;
   int    expFactor;
   int    mirror;
   double userScalar;
   /*** PAD ************/
   char   measPad[44];
   /*** DATA OFFSETS ***/
   int    numBytes;
   int    statusOffset;
   int    statusBytes;
   int    timeStampOffset;
   int    timeStampBytes;
   int    spectrumOffset;
   int    spectrumBytes;
   int    reserved_1;
   int    reserved_2;
   int    reserved_3;
   int    reserved_4;
   int    histogramOffset;
   int    histogramBytes;
   int    histoCountOffset;
   int    histoCountBytes;
   int    realTimeOffset;
   int    realTimeBytes;
   int    imagTimeOffset;
   int    imagTimeBytes;
   char   filePad[60];
};
```

Parameter Definitions

The following is a description of each parameter field:

General Information Section

version	This field contains the current version string of the data header. The current version is A.02.00. This version string changes if the header is no longer compatible with previous definitions of the header. The software always processes previous versions of the header but may not be able to process future versions.
valid	When this field is set, the associated data is valid. When it is not set (zero), the data is invalid.

Frequency Parameters

numLines	Specifies the number of values of frequency-domain data. The values are stored as magnitude (V^2) in 32-bit, floating precision. The current range of this field is from 51 to 12,801 lines
startFreq	Specifies the frequency value of the first frequency point.
deltaF	Specifies the frequency change between each value stored. To calculate the frequency value of any point, use the following equation: Fvalue = startFreq + (index * deltaF) where the index is the offset into the data block.

Time Parameters

blocksize	Specifies the number of points of time-domain data. The points are stored as voltage values in 32-bit, floating precision. The current range of this field from 128 points to 32,768 points.
startTime	Specifies the time value of the first time point.
deltaT	Specifies the time change between each point stored. To calculate the time value of any point, use the following equation: Tvalue = startTime + (index * deltaT) where the index is the offset into the data block.

Histogram Parameters

histBlkSize	Specifies the number of points of amplitude-domain data. The points are stored as counts in 32-bit floating precision. The current range of this field is from 51 points to 1,601 points.
startVoltage	Specifies the voltage value of the first amplitude point.
deltaV	Specifies the voltage change between each point stored. To calculate the voltage value of any point, use the following equation: Vvalue = startVoltage + (index * deltaV) where index is the offset into the data block.

Measurement Parameters

span	Specifies the frequency span of the data.

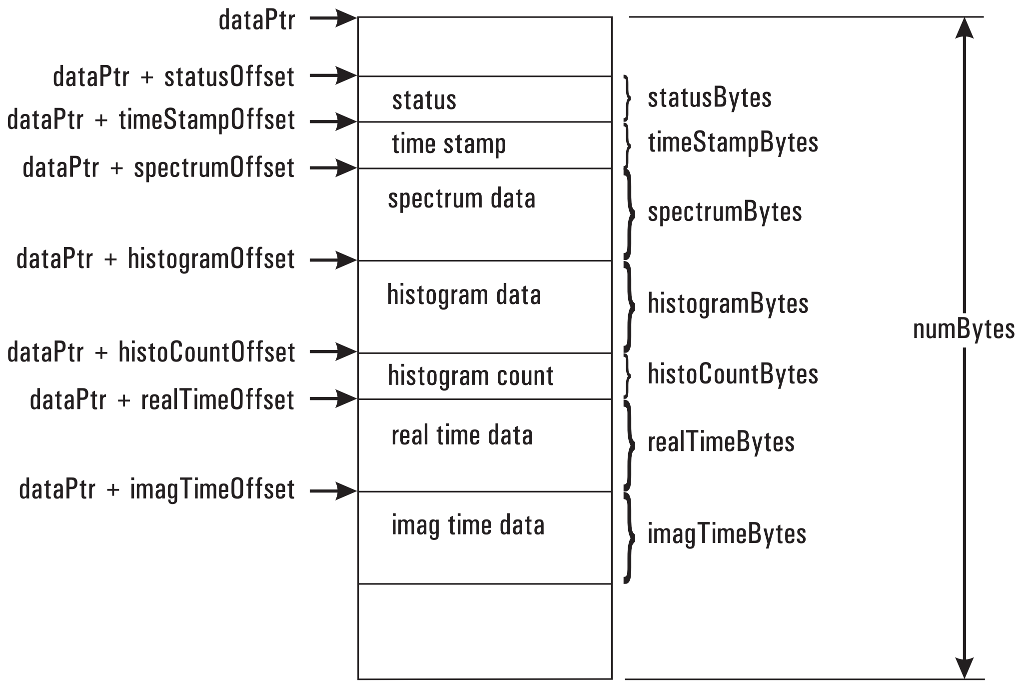
centerFreq	Specifies the center frequency of the data.
timeLength	Specifies the time length of the measurement block.
zoom	Specifies whether the data was mixed with a local oscillator or is a baseband measurement.
inputVoltage	Specifies the clipping level of the ADC in units of volts.
inputRange	Specifies the clipping level of the ADC in units of dBm.
numAverages	Specifies the number of averages.
overlap	Specifies the overlap percentage.
measResults	A bit field specifying the measurement results computed. The possible values are: DH_SPECTRUM_MR DH_REAL_TIME_MR DH_IMAG_TIME_MR DH_HISTOGRAM_MR
fs	Specifies the sample clocks frequency.
trigLevel	Specifies the trigger level in percent of clipping level.
trigMagLevel	Specifies the magnitude trigger level in units of dBFS/
trigDelay	Specifies the trigger delay in units of samples.
bandwidth	Specifies the filter bandwidth used. The possible values are: FULL_BANDWIDTH HALF_BANDWIDTH
coupling	Specifies the input coupling of the ADC. The possible values are: DC_COUPLED AC_COUPLED
analogFilterBypass	Specifies the analog anti-alias filter bypass mode. The possible values are: ANALOG_BYPASS_OFF ANALOG_BYPASS_ON
inputSource	Specifies the input source of the ADC. The possible values are: INPUT_SOURCE_SINGLE INPUT_SOURCE_DIFF
dataPrecision	Specifies the data precision of the ADC samples. The possible values are: DATA_PRECISION_16 DATA_PRECISION_32
trigSlope	Specifies the trigger slope of the ADC. The possible values are: TRIG_SLOPE_POS TRIG_SLOPE_NEG

trigMode	Specifies the trigger mode. The possible values are: TRIG_MODE_FREE TRIG_MODE_LEVEL TRIG_MODE_MAGNITUDE TRIG_MODE_EXT
windowType	Specifies the window type used with the FFT. The possible values are: WINDOW_TYPE_UNIFORM WINDOW_TYPE_HANN WINDOW_TYPE_FLATTOP WINDOW_TYPE_GAUSSTOP WINDOW_TYPE_BLACKMAN WINDOW_TYPE_GAUSSIAN
averageMode	Specifies the average mode used. The possible values are: AVERAGE_MODE_OFF AVERAGE_MODE_RMS AVERAGE_MODE_PEAK AVERAGE_MODE_EXP AVERAGE_MODE_NTH
expFactor	Specifies the exponential averaging factor.
mirror	Specifies whether the data is mirrored. The possible values are: MIRROR_OFF MIRROR_ON
userScalar	Specifies the additional scaling factor used.

Data Offsets

The data offsets in the header are the offsets of each field relative to the passed data pointer (dataPtr). In display functions, the data pointer is a passed parameter. In marker functions, the data pointer is an element in the aMarkerFuncParm structure. In stored data files, the data immediately follows the header in the file.

The following illustration shows the format of the data segment. The segments are stored in a file as discussed later in this chapter. If any offset values are less than zero, there is no data of that type stored in the data segment. The byte count for that field will be zero.



Data Structure for a Stored Data Segment

Data Fields

Status Field

The status field in version A.02.00 is a 32-bit integer. This field is used to store overload and gap information. If the STATUS_GAP bit is set, there was data lost between this data block and the previous block. If the STATUS_OVERLOAD bit is set, this data block contains data that was greater than the clipping level of the ADC.

Timestamp Field

The timestamp field in version A.02.00 is a 64-bit, floating-point number. It stores the time of acquisition of this data segment relative to the beginning of the measurement. The measurement begins when the RUN key is pressed.

Spectrum, Histogram, Real Time, and Imag Time Fields

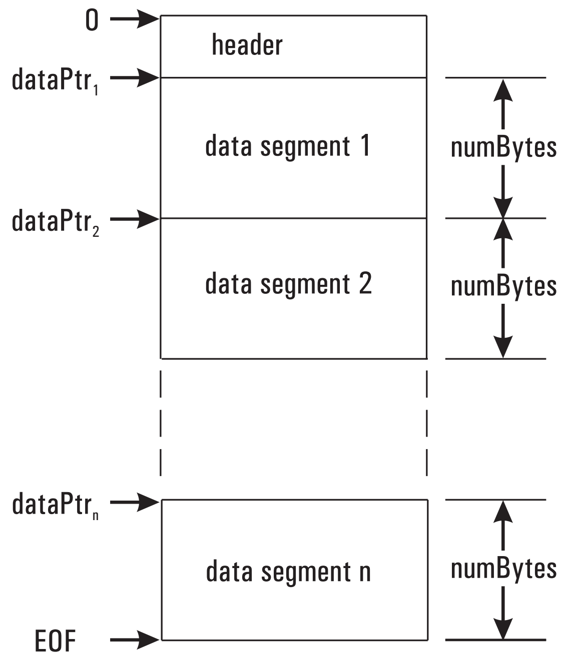
These field store their associated data as 32-bit, floating-point numbers. The spectrum field stores the data as magnitude-squared voltage values, the histogram field stores data as counts, and the time fields are stored as voltage values.

Histocount Field

The histocount field is a 32-bit, floating-point number which stores the total count of all the histogram bins. This is used to help compute the PDF and PDF display types.

Accessing Data Files on the Disk

When the HP 3587S software saves or records data to the disk, it first writes the header to the disk and then appends a data segment to the end of the file for each new measurement. To access the first data segment, you would lseek on the disk sizeof(struct aDataHeader) bytes into the file. This is the first dataPtr value. To go to the next data segment, add the numBytes value that is stored in the data header to the current value of dataPtr. You can continue through all the data segments in the file until you get the EOF error message.



Structure of a Data File

14

Creating Display Functions

This chapter explains how to create custom display functions using C subroutines. The dispType.h file contains the definition of the display structures used to implement display functions. The HP 3587s software allows you to create and/or customize as many as 25 different display types.

Display Types

A display type consists of 4 pieces of information.

1 Data type
2 Graph type
3 Display function
4 Scaling information

The data type field defines the measurement data used. The data types available are frequency, time, or amplitude domain data.

The graph type field picks the type of graphic presentation the data is displayed on. The graph types available are trace, map, color map, spectrogram, strip chart, digital persistence, and rollogram.

The display function is a routine that allows you to modify the data.

The scaling information lets the HP 3587s software know about the valid scaling information.

The following table shows where the currently defined display types are, relative to their data type and graph type.

Data Types versus Graph Types

Graph Types	**Data Types**		
	DATA_TYPE_FREQ	**DATA_TYPE_TIME**	**DATA_TYPE_HIST**
GRAPH_TYPE_TRACE	Spectrum	Time	Histogram, PDF, CDF
GRAPH_TYPE_MAP	Spectral Map	Time Map	Hist Map
GRAPH_TYPE_COLOR_MAP	Color Map		Hist Map
GRAPH_TYPE_SPECTROGRAM	Spectrogram		
GRAPH_TYPE_STRIP_CHART		Strip Chart	
GRAPH_TYPE_PERSIST	PR Spectrum	PR Time	
GRAPH_TYPE_ROLLOGRAM	Rollogram		

The HISTOGRAM, PDF, and CDF all have the same data and graph types but each one has a different display function to create the display type. See dispType.c for the code which computes the CDF and PDF results from histogram data.

Display Manager Structure

The aDispTypeManager structure consists of two elements: numDispTypes and an array of display type descriptions. The maximum number of display types is 25. When an additional display type description is defined, be sure to increment the numDispTypes field.

The following code is in dispType.h:

```
struct aDispTypeManager{
   int numDispTypes;
   struct aDispTypeDesc dtDesc[MAX_DISP_TYPES];
};
```

Display Description Structure

The aDispTypeDesc structure describes various attributes of each display type. The following code is in dispType.h:

```
struct aDispTypeDesc{
   char *name;
   int  dataType;
   void (*dispfunc)();
   int  graphType;
   int  family;
   char *helpText;
   struct aScaleDesc *scaleInfo[NUM_Y_COOR];
};
```

The following is a description of each field:

name — The name of the display type to appear in the user interface. It should be a maximum of 12 characters in length.

dataType — Selects which data type to assign to this display type. The choices are:
- DATA_TYPE_FREQ
- DATA_TYPE_TIME
- DATA_TYPE_HISTOGRAM
- DATA_TYPE_NONE

dispfunc — A pointer to a function that performs additional processing on incoming data. See Display Function later in this chapter for more information.

graphType — Selects which graph type to assign to this display type. The choices are:
- GRAPH_TYPE_TRACE
- GRAPH_TYPE_MAP
- GRAPH_TYPE_COLOR_MAP
- GRAPH_TYPE_SPECTROGRAM
- GRAPH_TYPE_STRIP_CHART
- GRAPH_TYPE_NONE
- GRAPH_TYPE_PERSIST
- GRAPH_TYPE_ROLLOGRAM

family — Used to group display types that are similar. This field is used when a new display type is chosen and you need to determine whether to reset the trace scaling information or keep the current setup. If you change from a SPECTRUM display type to a SPECTRAL MAP display type, you don't want to change the X-axis and Y-axis scale settings, If you change from a SPECTRAL MAP to a TIME display type, you should reset the scaling information.

helpText — Specifies the text for this display type when help information is requested.

scaleInfo — Specifies scaling information for this display type for each coordinate type. See Scaling Structure in the following discussion for more information on this structure.

Scaling Description Structure

Each display type has an aScaleDesc structure for each coordinate to define the scaling information needed in the HP 3587s software. The available coordinates are:

- Y_COOR_MAG_DBM
- Y_COOR_MAG_DB
- Y_COOR_MAG_LOG
- Y_COOR_MAG_LIN
- Y_COOR_REAL
- Y_COOR_IMAG
- Y_COOR_NONE

Not all coordinates are available for each data type. The next table shows which combinations are supported.

Data Types versus Supported Y-Coordinate Types

Y-Coordinate Types	Data Types		
	DATA_TYPE_FREQ	DATA_TYPE_TIME	DATA_TYPE_HIST
Y_COOR_MAG_DBM	X		
Y_COOR_MAG_DB	X		
Y_COOR_MAG_LOG	X		
Y_COOR_MAG_LIN	X		
Y_COOR_REAL		X	
Y_COOR_IMAG		X	
Y_COOR_NONE	X	X	X

The following code is in dispType.h:

```
struct aScaleDesc{
   int    yRefMode;
   int    yRangeMode;
   double maxYref;
   double minYref;
   double defaultYref;
   int    defaultYRangeIndex;
   double upperLimit;
   double lowerLimit;
   char   noneCoordLabel[16];
   double yRange[MAX_YRANGES];
};
```

The following is a description of each field:

yRefMode	Specifies the y-axis reference mode. The choices are : Y_REF_MODE_UNDEFINED Y_REF_MODE_TOP Y_REF_MODE_CENTER Y_REF_MODE_BOTTOM The Y reference is the location on the trace, top, bottom, or center, where the user interface value for the Y reference is assigned. Typically, frequency domain data has the reference at the top, time domain data in the center, and amplitude domain data at the bottom.
yRangeMode	Specifies the Y-axis range mode. The choices are: Y_RANGE_LIN Y_RANGE_LOG Y_RANGE_LIN specifies that the Y-axis is scaled linearly between the minimum and maximum Y-axis limits. Y_RANGE_LOG specifies that the Y-axis is scaled logarithmically between the minimum and maximum Y-axis limits.
maxYref	Specifies the maximum value the user can enter for the Y-reference value.
minYref	Specifies the minimum value the user can enter for the Y-reference value.
defaultYref	Specifies the initial Y-axis reference value.
defaultYRangeIndex	Specifies the index into the yRange array for the initial Y-axis range value.
upperLimit	Specifies the absolute maximum value for the Y-axis.
lowerLimit	Specifies the absolute minimum value for the Y-axis.
noneCoordLabel	Specifies the string to label the Y-axis with when the coordinate specified is none.
yRange	Array of Y-axis range values. Any Y-axis range value can be entered by the user as long as the upperLimit and lowerLimit values are not exceeded. This array is used by the knob entry to step in appropriate range steps like a 1-2-5 sequence.

Display Function

The display function is used to modify the data between the data buffer and the display. when data is acquired, it is stored in a data FIFO. the depth of this FIFO is specified by the parameter, buffer depth. The display software takes the data in this FIFO, scale it, and draw it to the screen. If a display function is specified, it is put in the processing chain between the data FIFO and the scaling routine. This routine is called anytime new data is required; if the user rescales a paused spectrogram display with 100 traces, the display function is called 100 times with each block of data. The direction in which the 100 traces are passed to the display is not guaranteed. Depending on the scroll direction, it could be from the newer data to older data or older data to newer data.

If a display function is going to be used to trigger an external device or only requires 'live' data, the mode bit should be checked to see if the DISP_FUNC_MODE_LIVE_DATA bit is set.

The calling convention for a display function is shown in the following:

```
void func(int mode, struct aDataHeader *header, char *data, float *output)
```

mode	If the DISP_FUNC_MODE_LIVE_DATA bit of this mode parameter is set, then it specifies that the data is coming live from the ADC module and not from a screen repaint request, playback of a signal capture file, or playback of a data file.
header	Pointer to the current data header. See header.h for more information.
data	Pointer to the incoming data. currently, this is all 32-bit, floating-point data.
output	Pointer to the location to which to write the output data. Exactly the same number of points valid in the input data needs to be written to this output pointer. These points must be 32-bit, floating-point numbers.

Display Function Example: demod.c

```
/************************************************************************
* demod.c
*
* This file implements AM, FM, and PM demodulation along with PHASE via the
* display function customization functionality of the HP3587s software.
*
************************************************************************
/
#include <stdlib.h>
#include <math.h>
#include "dispType.h"
#include "header.h"
#include "demod.h"

/************************************************************************
* Add the following lines to the end of the 'aDispTypeManager' structure
* and increment the display type count by 7 in the file 'dispType.c'.
*
"PHASE",
   DATA_TYPE_TIME, phase , GRAPH_TYPE_TRACE, 6, HELP_PHASE,
   NULL, NULL, NULL, NULL, NULL, NULL, &yPhaseRange,

"FM",
   DATA_TYPE_TIME, fm, GRAPH_TYPE_TRACE, 7, HELP_FM,
   NULL, NULL, NULL, NULL, NULL, NULL, &yFMRange,
"FM_STRIP",
   DATA_TYPE_TIME, fm, GRAPH_TYPE_STRIP_CHART, 7, HELP_FM_STRIP,
   NULL, NULL, NULL, NULL, NULL, NULL, &yFMRange,

"FM_MAP",
   DATA_TYPE_TIME, fm, GRAPH_TYPE_MAP, 7, HELP_FM_MAP,
   NULL, NULL, NULL, NULL, NULL, NULL, &yFMRange,

"AM",
   DATA_TYPE_TIME, am, GRAPH_TYPE_TRACE, 8, HELP_AM,
   NULL, NULL, NULL, NULL, NULL, NULL, &yAMRange,

"AM_STRIP",
   DATA_TYPE_TIME, am, GRAPH_TYPE_STRIP_CHART, 8, HELP_AM_STRIP,
   NULL, NULL, NULL, NULL, NULL, NULL, &yAMRange,

"PM",
   DATA_TYPE_TIME, pm, GRAPH_TYPE_TRACE, 9, HELP_PM,
   NULL, NULL, NULL, NULL, NULL, NULL, &yPMRange,
***********************************************************************/

/************************************************************************
* The scaling structures are formatted as shown below. For a complete
* definition of each of the fields, see the dispType.h include file.
*
*  struct aScaleDesc <name> = {
*      <yRefMode>, <yRangeMode>, <maxYref>, <minYref>, <defaultYref>,
*      <defaultYRangeIndex>, <upperLimit>, <lowerLimit>, <noneCoordLabel>,
*      <yRange[ 0 -  5]>,
*      <yRange[ 6 - 11]>,
*      <yRange[12 - 17]>,
*      <yRange[18 - 23]>
*  };
*
***********************************************************************/
```

```
struct aScaleDesc yPhaseRange = {
       Y_REF_MODE_CENTER, Y_RANGE_LIN, 180.0, -180.0, 0.0,
       21, 360.0, -360.0, "degrees",
       0.0002,   0.0004,   0.0010,   0.002,   0.004,   0.010,
         0.02,     0.04,     0.10,      0.2,     0.4,     1.0,
          2.0,      4.0,     10.0,     20.0,    40.0,    90.0,
        100.0,    200.0,    270.0,    360.0,   400.0,   720.0
};

struct aScaleDesc yFMRange = {
       Y_REF_MODE_CENTER, Y_RANGE_LIN, 4000000.0, -4000000.0, 0.0,
       16, 4000000.0, -4000000.0, "Hz",
       0.01, 0.2, 0.4, 1.0, 2.0, 4.0,
       10.0, 20.0, 40.0, 100.0, 200.0, 400.0,
       1000.0, 2000.0, 4000.0, 10000.0, 20000.0, 40000.0,
       100000.0, 200000.0, 400000.0, 1000000.0, 2000000.0, 4000000.0
};

struct aScaleDesc yAMRange = {
       Y_REF_MODE_BOTTOM, Y_RANGE_LIN, 10.0, 0.0, 0.0,
       18, 100.0, 0.0, "",
       0.000001,0.000002,0.000005, 0.00001,0.00002,0.00005,
       0.0001,0.0002,0.0005, 0.001,0.002,0.005,
       0.01,0.02,0.05, 0.1,0.2,0.5,
       1.0,2.0,5.0, 10.0,20.0,50.0
};

struct aScaleDesc yPMRange = {
       Y_REF_MODE_CENTER, Y_RANGE_LIN, 5040.0, -5040.0, 0.0,
       9, 5040.0, -5040.0, "degrees",
       2.0,      4.0,     10.0,     20.0,    40.0,    90.0,
     100.0,    200.0,    270.0,    360.0,   400.0,   720.0,
    1080.0,   1440.0,   1800.0,   2160.0,  2520.0,  2880.0,
    3240.0,   3600.0,   3960.0,   4320.0,  4680.0,  5040.0
};

/*************************************************************************
* phase
*
* This function computes the phase versus time of the incoming signal. This
* is accomplished by taking the arctangent of imaginary part divided by the
* real part of the signal. If the signal is baseband, ie.only the real part
* then the phase will be 180 degrees if greater than zero, or -180 degrees
* if less than zero. If no time data is available, the output data is zero.
*
*************************************************************************/
void phase(int mode, struct aDataHeader *header, char *data, float *output)
{
   float *rptr, *iptr;
   double real, imag, scalar;
   int i;

   if(header->measResults & (DH_IMAG_TIME_MR | DH_REAL_TIME_MR)){
      rptr = (float *)(data + header->realTimeOffset);
      iptr = (float *)(data + header->imagTimeOffset);
      scalar = 180.0/M_PI;
      for(i=0;i<header->blocksize;i++){
         real = *rptr++;
         imag = *iptr++;
         *output++ = (float)(scalar * atan2(imag,real));
      }
   }
   else if(header->measResults & DH_REAL_TIME_MR){
      rptr = (float *)(data + header->realTimeOffset);
      for(i=0;i<header->blocksize;i++){
```

```
            real = *rptr++;
            if(real > 0.0)
               *output++ = 180.0;
            else if(real < 0.0)
               *output++ = -180.0;
            else
               *output++ = 0.0;
         }
      }
      else{
         for(i=0;i<header->blocksize;i++){
            *output++ = 0.0;
         }
      }
}
/***************************************************************************
* fm
*
* This function computes the frequency modulation versus time of the
* incoming signal.
*
***************************************************************************/
void fm(int mode, struct aDataHeader *header, char *data, float *output)
{
   float *rptr, *iptr, oldPhase, phase, phaseRef, delta, freqScalar;
   int i, step;
   float phaseArray[32768];

   if(header->measResults & (DH_IMAG_TIME_MR | DH_REAL_TIME_MR)){
      rptr = (float *)(data + header->realTimeOffset);
      iptr = (float *)(data + header->imagTimeOffset);
      /******************************************
      * Calculate the derivative of the phase  *
      * of the incoming signal.                *
      ******************************************/
      phaseRef = 0.0;
      phase = atan2(*iptr++,*rptr++);
      phaseArray[0] = phaseRef + phase;
      for(i=1;i<header->blocksize;i++){
         oldPhase = phase;
         phase = atan2(*iptr++,*rptr++);
         delta = phase - oldPhase;
         if(delta > M_PI) phaseRef -= 2*M_PI;
         else if(delta < -M_PI) phaseRef += 2*M_PI;
         phaseArray[i] = phaseRef + phase;
      }
      /*******************************************
      * Compute the FM demod by calculating the *
      * derivative of the PM. Smooth the result *
      * by taking the phase difference between  *
      * 'step' points and then dividing by      *
      * 'step'. The first 'step' points are     *
      * handled differently. They are computed  *
      * by taking the current difference until  *
      * step points have been calculated.       *
      *******************************************/
      step = 8;
      freqScalar = 1.0/(2.0 * M_PI * header->deltaT);
      if(header->mirror) freqScalar = -freqScalar;
      iptr = rptr = &phaseArray[0];
      iptr++;
      for(i=1;i<step;i++){
         delta = (*iptr++ - *rptr) * freqScalar/(double)i;
         if(i == 1){
```

```
            *output++ = delta;
            *output++ = delta;
         }
         *output++ = delta;
      }
      freqScalar *= 1.0/(double)step;
      for(i=step;i<header->blocksize;i++){
         *output++ = (*iptr++ - *rptr++) * freqScalar;
      }
   }
   else{
      for(i=0;i<header->blocksize;i++){
         *output++ = 0.0;
      }
   }
}

/***************************************************************************
* am
*
* This function computes the amplitude modulation versus time of the
* incoming signal. The amplitude modulation is the square root of the sum
* of the squares of the real and imaginary parts of the complex input time
* data.
*
**************************************************************************/
void am(int mode, struct aDataHeader *header, char *data, float *output)
{
   float *rptr, *iptr;
   double real, imag;
   int i;

   if(header->measResults & (DH_IMAG_TIME_MR | DH_REAL_TIME_MR)){
      rptr = (float *)(data + header->realTimeOffset);
      iptr = (float *)(data + header->imagTimeOffset);

      for(i=0;i<header->blocksize;i++){
         real = *rptr++;
         imag = *iptr++;
         *output++ = (float)(sqrt(real*real + imag*imag));
      }
   }
   else{
      for(i=0;i<header->blocksize;i++){
         *output++ = 0.0;
      }
   }
}

/***************************************************************************
* pm
*
* This function computes the phase modulation versus time of the incoming
* signal. This is done by taking the derivative of the phase of the time
* signal.
*
**************************************************************************/
void pm(int mode, struct aDataHeader *header, char *data, float *output)
{
   float *rptr, *iptr;
   double oldPhase, phase, phaseRef, delta, real, imag;
   int i;

   if(header->measResults & (DH_IMAG_TIME_MR | DH_REAL_TIME_MR)){
      rptr = (float *)(data + header->realTimeOffset);
```

```
      iptr = (float *)(data + header->imagTimeOffset);
      phaseRef = 0.0;
      real = *rptr++;
      imag = *iptr++;
      phase = atan2(imag,real);
      *output++ = phaseRef + phase;
      for(i=1;i<header->blocksize;i++){
         oldPhase = phase;
         real = *rptr++;
         imag = *iptr++;
         phase = atan2(imag,real);
         delta = phase - oldPhase;
         if(delta > M_PI) phaseRef -= 2*M_PI;
         if(delta < -M_PI) phaseRef += 2*M_PI;
         *output++ = phaseRef + phase;
      }
   }
   else{
      for(i=0;i<header->blocksize;i++){
         *output++ = 0.0;
      }
   }
}
```

To Modify Display and Marker Functions

If you wish to modify the display and marker functions, you should copy the entire /usr/hp3587s/src/functions directory to your user area such as /users/hp3587s/functions. To do this, perform the following steps:

1. `cd /users/hp3587s <Return>`
2. `mkdir functions <Return>`
3. `cd functions <Return>`
4. `cp /usr/hp3587s/src/functions/* . <Return>`
5. to recompile the code, type `make <Return>`

The make file creates the file `hp3587s`. To replace this file with the one shipped, perform the following steps:

For HP-UX 9.0

1. `su <Return>` Become super-user
2. `mv /usr/bin/hp3587s /usr/bin/hp3587s.orig <Return>` Save original
3. `cp $HOME/functions/hp3587s /usr/bin/hp3587s <Return>`

For HP-UX 10.20

1. su <Return> Become super-user
2. `mv /opt/hp3587s/bin/hp3587s /opt/hp3587s/bin/hp3587s.orig <Return>` Save original
3. `cp $HOME/functions/hp3587s /opt/hp3587s/bin/hp3587s <Return>`

To run the analyzer, type `hp3587s <Return>` from a window command line.

15

Creating Marker Functions

This chapter explains how to create custom marker functions using C subroutines. The markFunc.h file contains the definition of the marker function parameter structure which is passed to the user-defined marker function routines.

See also, “To Modify Display and Marker Functions” in chapter 14 for a discussion of make files.

Marker Function Parameters

A marker function routine is passed two parameters: a mode and the aMarkerFuncParm structure. The mode is one of two values: MF_MODE_FILL or MF_MODE_EXECUTE.

When the marker function receives the MF_MODE_FILL mode, it should fill line1, line2, and helpText fields of the passed aMarkerFuncParm structure. The line1 and line2 fields are used to label the function key in the user interface. The help text is displayed when the function key is pressed along with the <Shift> key. The help text should not be longer than 256 characters.

When the marker function receives the MF_MODE_EXECUTE mode, it should perform its marker function and do any or all of the following:

- move the marker
- move the relative marker
- return a string to be displayed

The Marker Function Structure

```
struct aMarkerFuncParm{
   int    dataType;
   int    graphType;
   struct aDataHeader *dh;
   int    startBin;
   int    stopBin;
   int    markerMode;
   double yMin;
   double yMax;
   int    peak_or_rms;
   int    volt_or_volt2;
   char   xUnit[MFP_X_UNIT_LENGTH];
   char   yUnit[MFP_Y_UNIT_LENGTH];
   int    yCoordinate;
   int    yRefMode;
   int    markerBin;
   float  *data;
   double time;
   int    relativeBin;
   float  *rel_data;
   double rel_time;
   char   line1[MFP_LINE1_LENGTH];
   char   line2[MFP_LINE2_LENGTH];
   char   *helpText;
   int    action;
   char   returnString[MFP_RETURN_STRING_LENGTH];
   int    returnMarkerBin;
   int    returnRelativeBin;
};

void getWindowWideCorrection(int, double *);
void markerFunc1(int, struct aMarkerFuncParm *);
void markerFunc2(int, struct aMarkerFuncParm *);
void markerFunc3(int, struct aMarkerFuncParm *);
void markerFunc4(int, struct aMarkerFuncParm *);
void markerFunc5(int, struct aMarkerFuncParm *);
void markerFunc6(int, struct aMarkerFuncParm *);
void markerFunc7(int, struct aMarkerFuncParm *);
void markerFunc8(int, struct aMarkerFuncParm *);
void markerFunc9(int, struct aMarkerFuncParm *);
void markerFunc10(int, struct aMarkerFuncParm *);
```

Marker Function Structure Field Definitions

dataType	Specifies the current trace data type. It can be one of the following values: MFP_DATA_TYPE_FREQ MFP_DATA_TYPE_TIME MFP_DATA_TYPE_HISTOGRAM MFP_FATA_TYPE_NONE
graphType	Specifies the current graph type. It can be one of the following values: MFP_GRAPH_TYPE_TRACE MFP_GRAPH_TYPE_MAP MFP_GRAPH_TYPE_COLOR_MAP MFP_GRAPH_TYPE_SPECTROGRAM MFP_GRAPH_TYPE_STRIP_CHART MFP_GRAPH_TYPE_NONE
dh	Pointer to the data header. See header.h for more information on this structure.
startBin	The first bin in the data block that is displayed on the trace.
stopBin	The last bin in the data block that is displayed on the trace.
markerMode	The current marker mode. The possible values are: MFP_MARKER_MODE_OFF MFP_MARKER_MODE_SINGLE MFP_MARKER_MODE_RELATIVE
yMin	The minimum Y-axis value of the trace.
yMax	The maximum Y-axis value of the trace.
peak_or_rms	Specifies the user selection to display peak or rms voltage data. The possible values are MFP_UNIT_PEAK MFP_UNIT_RMS
volt_or_volt2	Specifies the user selection to display voltage or voltage-squared data. Possible values are: MFP_UNIT_NOT_POWER MFP_UNIT_POWER
xUnit	The X-axis unit string.
yUnit	The Y-axis unit string.

yCoordinate	The Y-axis trace coordinate. The possible values are: MFP_Y_COOR_MAG_DBM MFP_Y_COOR_MAG_DB MFP_Y_COOR_MAG_LOG MFP_Y_COOR_MAG_LIN MFP_Y_COOR_REAL MFP_Y_COOR_IMAG MFP_Y_COOR_NONE
yRefMode	The Y-axis reference mode. the possible values are: MFP_Y_REF_MODE_UNDEFINED MFP_Y_REF_MODE_TOP MFP_Y_REF_MODE_CENTER MFP_Y_REF_MODE_BOTTOM
markerBin	The data bin on which the marker is currently located.
data	Pointer to the 32-bit, floating-point data block on which the marker is currently located.
time	The time stamp value of the data block on which the marker is currently located.
relativeBin	The data bin in which the relative marker is currently located.
rel_data	Pointer to the 32-bit, floating-point data block on which the relative marker is currently located.
rel_time	The time stamp value of the data block on which the relative marker is currently located.
line1	Field to return the top line of information to display on the user interface function key. The maximum length of this string is MFP_LINE1_LENGTH, including the terminating '\0'character.
line2	Field to return the bottom line of information to display on the user interface function key. The maximum length of this string is MFP_LINE2_LENGTH, including the terminating '\0' character.
helpText	Specifies the pointer to the marker function help text which is displayed in the user interface. The help text must be global so that it is available outside of the marker function
action	Specifies the actions you want the user interface to perform upon return from this marker function. The possible values are: MF_ACTION_NONE — do nothing MF_ACTION_DISPLAY_STRING — display string in returnString field MF_ACTION_MOVE_MARKER — move marker to value in returnMarkerBin MF_ACTION_MOVE_REL_MARKER — move relative marker to returnRealtiveBin All these actions can be ORed together.

Marker Function Example: intermod.c

```
/***************************************************************************
* intermod.c
*
*  This file implements a marker function to calculate the 3rd order
*  intermodulation distortion of a two tone signal. To add this marker
*  function to the hp3587s software, comment out the 'markerFunc6' function
*  in markFunc.c, and add this file into the makefile.
*
***************************************************************************/
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include "header.h"
#include "markFunc.h"

#define HELP_INTERMOD "~Computes the 3rd order intermodulation distortion
level resulting from a two tone input source."

void markerFunc6(int mode, struct aMarkerFuncParm *mfp)
{
   char   string[80];
   float  max1, max2, *ptr, point, startFreq, stopFreq;
   float  freq1, freq2, fintermod;
   int    i, order, maxBin1, maxBin2, intermodBin, numPoints, m1, m2,
found;
   int    m1Max, m2Max, intermodBinMax;
   double result, fullScale, intermodValue, intermodMax;

   switch(mode){
      case MF_MODE_FILL:
         strcpy(mfp->line1,"3rd ORDER");
         strcpy(mfp->line2,"INTERMOD");
         mfp->helpText = HELP_INTERMOD;
         break;

      case MF_MODE_EXECUTE:
         if(mfp->dataType != MFP_DATA_TYPE_FREQ){
            strcpy(mfp->returnString,"WRONG DATA TYPE");
            mfp->action = MF_ACTION_DISPLAY_STRING;
            break;
         }
         if(mfp->markerMode != MFP_MARKER_MODE_RELATIVE){
            strcpy(mfp->returnString,"WRONG MARKER MODE");
            mfp->action = MF_ACTION_DISPLAY_STRING;
            break;
         }
         /*****************************************/
         /* find the highest two peaks            */
         /*****************************************/
         numPoints = mfp->stopBin - mfp->startBin + 1;
         maxBin1 = maxBin2 = mfp->startBin;
         ptr = &mfp->data[maxBin1];
         max1 = max2 = *ptr++;
         for(i=1;i<numPoints;i++){
            point = *ptr++;
            if(point > max1){
               max2 = max1;
               maxBin2 = maxBin1;
               max1 = point;
               maxBin1 = mfp->startBin + i;
```

```
            }
            else if(point > max2){
               max2 = point;
               maxBin2 = mfp->startBin + i;
            }
         }
         freq1 = mfp->dh->startFreq + (maxBin1 * mfp->dh->deltaF);
         freq2 = mfp->dh->startFreq + (maxBin2 * mfp->dh->deltaF);
         /*******************************************/
         /* compute fullscale value                 */
         /*******************************************/
         if(max1 >= max2)
            fullScale = max1;
         else
            fullScale = max2;
         if(fullScale <= 0.0) fullScale = -600.0;
         else fullScale = 10.0 * log10(fullScale);
         /*******************************************/
         /* compute all intermods                   */
         /*******************************************/
         startFreq = mfp->dh->startFreq + (mfp->startBin * mfp->dh-
>deltaF);
         stopFreq  = mfp->dh->startFreq + (mfp->stopBin * mfp->dh->deltaF);
         order     = 3;
         found     = 0;
         intermodMax = -600.0;
         for(i=-order;i<=order;i++){
            m1 = i;
            m2 = order - abs(i);
            fintermod = (freq1 * (float)m1) + (freq2 * (float)m2);
            if(fintermod >= startFreq && fintermod <= stopFreq){
               found = 1;
               intermodBin = floor(((fintermod - mfp->dh->startFreq) /
                                             mfp->dh->deltaF) +
0.5);
               intermodValue = mfp->data[intermodBin];
               if(intermodValue <= 0.0) intermodValue = -600.0;
               else intermodValue = 10.0 * log10(intermodValue);
               result = intermodValue - fullScale;
               if(result > intermodMax){
                  intermodMax = result;
                  intermodBinMax = intermodBin;
                  m1Max = m1;
                  m2Max = m2;
               }
            }
            if(m2 != 0){
               m2 = -m2;
               fintermod = (freq1 * (float)m1) + (freq2 * (float)m2);
               if(fintermod >= startFreq && fintermod <= stopFreq){
                  found = 1;
                  intermodBin = floor(((fintermod - mfp->dh->startFreq) /
                                             mfp->dh->deltaF) +
0.5);
                  intermodValue = mfp->data[intermodBin];
                  if(intermodValue <= 0.0) intermodValue = -600.0;
                  else intermodValue = 10.0 * log10(intermodValue);
                  result = intermodValue - fullScale;
                  if(result > intermodMax){
                     intermodMax = result;
                     intermodBinMax = intermodBin;
                     m1Max = m1;
                     m2Max = m2;
                  }
```

```
            }
         }
      }

      /*****************************************/
      /* pass back results                     */
      /*****************************************/
      if(!found){
         strcpy(mfp->returnString,"NO INTERMODS IN BAND");
         mfp->action = MF_ACTION_DISPLAY_STRING;
      }
      else{
         sprintf(string,"3rd Order [%df1%+df2]: %.2f dBc",
                                                m1Max, m2Max,
intermodMax);
         strncpy(mfp->returnString,string,39);
         mfp->returnString[39] = '\0';
         mfp->returnMarkerBin = intermodBinMax;
         mfp->returnRelativeBin = maxBin1;
         mfp->action = MF_ACTION_DISPLAY_STRING |
                       MF_ACTION_MOVE_MARKER |
MF_ACTION_MOVE_REL_MARKER;
      }
      break;
   }
}
```

Index