ManualsPlus.com

INTERFACING AND PROGRAMMING MANUAL

HP 7550A
Graphics Plotter

©1984, 1986 by Hewlett-Packard Company
16399 W. Bernardo Drive, San Diego, CA 92127-1899

ManualsPlus.com

NOTICE

The information contained in this document is subject to change without notice.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

PRINTING HISTORY

First Edition — March 1984

Change Sheet to First Edition — July 1984

Second Edition — October 1984

Third Edition — January 1986

Fourth Edition — July 1986

ManualsPlus.com

How to Use the HP 7550 Documentation

All HP 7550 plotters are shipped with the following three documents.

- **Operation and Interconnection Manual** (Part No. 07550-90002). The Operation and Interconnection Manual contains detailed operating information such as how to load paper and pens, how to use the front-panel function keys, and how to obtain quality plots using proper pen/media combinations. It also describes how to connect your computer and plotter.
- **Interfacing and Programming Manual** (Part No. 07550-90001). This is the manual you are reading now. It contains complete explanations of the plotter's graphics and interfacing instructions. It begins with discussions of various programming languages and how to use them with the plotter's instruction set, HP-GL (Hewlett-Packard Graphics Language). It also explains some fundamental plotting concepts to help get you started.
- **Pocket Guide** (Part No. 07550-90003). The Pocket Guide is a convenient reference list of all HP-GL and device-control instructions, along with their parameters.

The following paragraphs describe how you can use these documents, based on your expertise.

For First Encounters with the HP 7550

If you have just received your plotter, read the first three chapters of the Operation and Interconnection Manual. These chapters describe initial inspection and explain the front-panel function keys. Next, look for your computer in Chapter 6 and verify communication between your computer and plotter. If you are using a purchased graphics software package, be sure to follow any special instructions provided in the documentation accompanying the software.

Now you can use your graphics software, or you can program the plotter yourself. If you plan to write programs for the plotter, read the following two paragraphs.

For First Encounters with HP-GL

If you have never written programs in HP-GL, begin with the first chapter of this manual. This chapter will help you decide whether to program with your computer's graphics statements/routines or with HP-GL. The second chapter explains some basic plotting concepts, and the remaining chapters present the HP-GL instructions. The examples given with the instructions will help you learn how the instructions work.

ManualsPlus.com

For Experienced HP-GL Programmers

If you are an experienced HP-GL programmer, you might find the Pocket Guide or the instruction summary in Appendix C of this manual most helpful. Also, look through this manual to learn about the new instructions not found in earlier plotters — for example, the polygon instructions, area-fill instructions, replot instructions, and expanded character sets and character-designing capabilities. If you are interested in the differences in syntax between this and other HP plotters, read Chapter 3 of this manual.

ManualsPlus.com

Table of Contents

How to Use the HP 7550 Documentation iii

Chapter 1: Introduction to Programming the Plotter 1-1
What You'll Learn in This Chapter 1-1
What Can Your HP 7550 Plotter Do for You? 1-1
Programming Languages — Which Should You Use? 1-5
HP-GL and Device-Control Instructions 1-5
BASIC 1-6
FORTRAN and Pascal 1-6
How to Use the Examples in This Manual 1-6
Examples Presented as Complete Programs 1-7
Examples Presented as HP-GL Strings 1-8
HP Series 80 and HP Series 200 Computers —
A Note To Users 1-9
Understanding Manual Conventions 1-10
Developing a Plot 1-10

Chapter 2: Fundamental Plotting Concepts 2-1
What You'll Learn in This Chapter 2-1
Introduction to the Cartesian Coordinate System 2-1
Plotting with the Coordinate System 2-2
Plotter Units 2-3
User Units 2-4
Scaling 2-5
Graphics Limits 2-9
Hard-Clip Limits 2-9
Soft-Clip Limits 2-10
Pen Movement 2-12
Current Pen Status and Pen Position 2-12
Absolute and Relative Movement 2-13

Chapter 3: Preliminary Setup Using HP-GL 3-1
What You'll Learn in This Chapter 3-1
HP-GL Instructions Covered 3-1
Terms You Should Understand 3-1
HP-GL Syntax 3-2
Omitting Optional Parameters 3-3
Parameter Formats 3-4
Notations Used in This Manual for Expressing Syntax 3-5
Compatibility with Syntax of Other HP-GL Plotters 3-5
A Note about HP-GL Errors 3-6
The Default Instruction, DF 3-8
The Initialize Instruction, IN 3-11
The Input P1 and P2 Instruction, IP 3-12

ManualsPlus.com

Chapter 3: Preliminary Setup Using HP-GL (Continued)
The Scale Instruction, SC 3-14
The Graphics Memory Instruction, GM 3-18
Hints for Using the GM Instruction 3-19

Chapter 4: Pen Control and Plotting 4-1
What You'll Learn in This Chapter 4-1
HP-GL Instructions Covered 4-1
Terms You Should Understand 4-1
The Select Pen Instruction, SP 4-2
The Pen Instructions, PU and PD 4-3
The Plot Absolute Instruction, PA 4-4
Plotting with PA, PU, and PD 4-5
The Plot Relative Instruction, PR 4-6
Plotting with PR, PU, and PD 4-8
Relationship of Plotting Instructions and Graphics Limits 4-9
Types of X,Y Coordinates 4-9
Plotting Inside and Outside of Windows 4-10
Lost Mode 4-11
Plotting with Variables 4-12
Methods for Sending Variable Parameters 4-12
Example — Plotting a Line Chart with Variable Parameters 4-13

Chapter 5: Enhancing Your Plots 5-1
What You'll Learn in This Chapter 5-1
HP-GL Instructions Covered 5-1
Terms You Should Understand 5-1
The Tick Instructions, XT and YT 5-1
Example — Drawing X-Ticks 5-2
The Tick Length Instruction, TL 5-2
Example — Drawing Grid Lines, Major Ticks, and Minor Ticks 5-3
The Symbol Mode Instruction, SM 5-5
Example — Plotting in Symbol Mode 5-6
The Line Type Instruction, LT 5-6
Example — Fixed and Adaptive Line Types 5-9

Chapter 6: Circles, Arcs, and Polygons 6-1
What You'll Learn in This Chapter 6-1
HP-GL Instructions Covered 6-1
Terms You Should Understand 6-2
The Chord Tolerance Instruction, CT 6-3

ManualsPlus.com

Table of Contents (Continued)

Chapter 6: Circles, Arcs, and Polygons (Continued)
The Circle Instruction, CI 6-5
Example — Effects of Chord Tolerance on Circle Smoothness 6-7
Example — Drawing Circles with Different Radii and Line Types 6-8
The Arc Instructions, AA and AR 6-9
Example — Using the AA Instruction 6-12
Example — Using the AR Instruction 6-12
The Fill Type Instruction, FT 6-13
Example — Effects of Line Type Patterns 6-15
The User-Defined Fill Type Instruction, UF 6-17
Examples — Creating Special Effects 6-18
The Pen Thickness Instruction, PT 6-20
The Wedge Instructions, WG and EW 6-21
Example — Defining and Filling Wedges Using the WG Instruction 6-24
Example — Outlining the Wedges Using the EW or EP Instructions 6-26
The Absolute Rectangle Instructions, RA and EA 6-27
Example — Defining and Filling Rectangles Using the RA Instruction 6-29
Example — Outlining the Rectangles Using the EA or EP Instructions 6-30
Example — Drawing an Organization Chart with the EA Instruction 6-32
The Relative Rectangle Instructions, RR and ER 6-33
Examples — Using Relative Coordinates to Draw Rectangles 6-34
The Polygon Mode Instruction, PM 6-37
Example — Creating Block Letters in Polygon Mode 6-40
Example — Using the CI Instruction in Polygon Mode 6-42
The Edge Polygon Instruction, EP 6-43
The Fill Polygon Instruction, FP 6-44
Example — Creating a Surface Chart 6-45
Example — Filling Alternating Subpolygons 6-47
The Polygon Buffer 6-48
Determining the Approximate Size of the Polygon Buffer 6-49
Counting the Points in Your Polygon 6-49
Determining the Exact Size of the Polygon Buffer 6-50

Chapter 7: Labeling Basics 7-1
What You'll Learn in This Chapter 7-1

ManualsPlus.com

Table of Contents (Continued)

Chapter 7: Labeling Basics (Continued)
HP-GL Instructions Covered 7-1
Terms You Should Understand 7-1
The Label Instruction, LB 7-2
Character Positioning 7-4
How to Send the Label Terminator and Other Nonprinting Characters 7-6
Example — Repeating Labels from the Label Buffer 7-7
The Define Label Terminator Instruction, DT 7-8
Example — Changing Label Terminators 7-9
Labeling with Variables 7-10
Adjusting Character Size, Spacing, and Position 7-11
The Character Plot Cell 7-12
Character Size 7-13
Direction and Slant 7-13
Spacing and Position 7-13
The Absolute Character Size Instruction, SI 7-14
Example — Changing Absolute Character Size 7-15
Example — Using Negative Parameters to Mirror Labels 7-15
The Relative Character Size Instruction, SR 7-16
Example — Changing Relative Character Size 7-17
The Character Slant Instruction, SL 7-18
Example — Specifying Character Slant 7-20
The Absolute Direction Instruction, DI 7-21
Examples — Rotating Label Direction 7-23
The Relative Direction Instruction, DR 7-25
Example — Effects of Changing P1 or P2 on Relative Label Direction 7-27
The Label Origin Instruction, LO 7-30
Examples — Positioning Labels with LO 7-32
The Character Plot Instruction, CP 7-33
Example — Using CP to Align Labels 7-35
The Extra Space Instruction, ES 7-36
The Buffer Label Instruction, BL 7-38
Example — Positioning Buffered Labels with the LO Instruction 7-39
The Print Buffered Label Instruction, PB 7-39
The Output Label Length Instruction, OL 7-40
Example — Underlining a Label Using Information from OL 7-41
Parameter Interaction in Labeling Instructions 7-42
Using DI and SI Together 7-42

ManualsPlus.com

Table of Contents (Continued)

Chapter 7: Labeling Basics (Continued)
Using DR and SI Together 7-43
Using DI and SR Together 7-44
Using DR and SR Together 7-46

Chapter 8: Putting the Instructions to Work 8-1
What You'll Learn in This Chapter 8-1
A Reminder about HP-GL Syntax 8-2
A Reminder about BASIC 8-2
Line Chart 8-2
Setup and Scaling 8-2
The Axes and Their Labels 8-3
Plotting the Data 8-6
Program Listing 8-7
Bar Charts and Pie Charts 8-9
Filling and Hatching 8-9
Producing a Bar Chart 8-10
Producing a Pie Chart 8-13

Chapter 9: Changing the Plotting Area 9-1
What You'll Learn in This Chapter 9-1
HP-GL Instructions Covered 9-1
Terms You Should Understand 9-1
The Rotate Coordinate System Instruction, RO 9-2
The Input Window Instruction, IW 9-5
Example — Effects of Specifying a Window on Labels and Lines 9-8
The Output Window Instruction, OW 9-9
The Output Hard-Clip Limits Instruction, OH 9-10
The Output P1 and P2 Instruction, OP 9-11
Techniques for Changing the Plotting Area 9-12
Preparing Equal-Sized Plots on One Page 9-12
Reducing/Enlarging Plots 9-13
Using Windows When Scaling Is On to Reduce/Enlarge Portions of a Plot 9-14
Creating Mirror Images 9-17

Chapter 10: Front-Panel Functions — Pen Control, Paper Feed, Replotting, and Function Keys 10-1
What You'll Learn in This Chapter 10-1
HP-GL Instructions Covered 10-1
Terms You Should Understand 10-2
The Automatic Pen Operations Instruction, AP 10-2

ManualsPlus.com

Table of Contents (Continued)

Chapter 10: Front-Panel Functions — Pen Control, Paper Feed, Replotting, and Function Keys (Continued)
The Group Pen Instruction, GP 10-4
The Select Pen Group Instruction, SG 10-6
The Force Select Instruction, FS 10-6
The Acceleration Select Instruction, AS 10-7
The Velocity Select Instruction, VS 10-8
The Curved Line Generator Instruction, CV 10-10
The Page Feed Instruction, PG (or AF or AH) 10-12
The Not-Ready Instruction, NR 10-14
The Buffer Plot Instruction, BF 10-15
The Replot Instruction, RP 10-16
The Interactive Front-Panel Display 10-17
The Write to Display Instruction, WD 10-18
The Define Key Instruction, KY 10-20
Example — Defining Function Keys with the KY Instruction 10-22
The Output Key Instruction, OK 10-23
Example — Using the Function Keys with the OK and WD Instructions 10-24
The Group Count Instruction, GC 10-25
The Output Group Count Instruction, OG 10-26

Chapter 11: Labeling with Alternate Character Sets and Designing Characters 11-1
What You'll Learn in This Chapter 11-1
HP-GL Instructions Covered 11-1
Terms You Should Understand 11-1
Plotter Character Sets 11-2
Character Selection Modes 11-4
Labeling with Standard and Alternate Character Sets 11-4
Selecting Sets with the SS and SA Instructions 11-5
Selecting Sets with the Shift-In/Shift-Out Control Characters 11-6
The Designate Standard Character Set Instruction, CS 11-7
The Designate Alternate Character Set Instruction, CA 11-8
The Select Standard Character Set Instruction, SS 11-9
The Select Alternate Character Set Instruction, SA 11-9
Choosing Other Character Selection Modes 11-10
The In-Use Code Table 11-11
Control Characters 11-13
Fallback Mode 11-14
HP 7-Bit Compatibility Mode 11-14

ManualsPlus.com

Table of Contents (Continued)

Chapter 11: Labeling with Alternate Character Sets and Designing Characters (Continued)
HP 8-Bit Mode 11-15
ISO 7-Bit Mode 11-16
Example — Using the DS and IV Instructions in the ISO 7-Bit Mode 11-17
ISO 8-Bit Mode 11-18
Example — Embedding a Single-Shift in a Label String 11-19
The Character Selection Mode Instruction, CM 11-20
The Designate Character Set into Slot Instruction, DS 11-22
The Invoke Character Slot Instruction, IV 11-24
The Character Chord Angle Instruction, CC 11-26
Example — Effects of Changing Character Chord Angles 11-27
The User-Defined Character Instruction, UC 11-28
Example — Defining Your Own Characters with the UC Instruction 11-34
The Define Downloadable Character Instruction, DL 11-36
Allocating Memory for Downloadable Characters 11-39
Example — Defining a Downloadable Character 11-40

Chapter 12: Digitizing 12-1
What You'll Learn in This Chapter 12-1
HP-GL Instructions Covered 12-1
Terms You Should Understand 12-1
Preparing Your Plotter for Use as a Digitizer 12-2
The Digitize Point Instruction, DP 12-2
The Output Digitized Point and Pen Status Instruction, OD 12-3
The Digitize Clear Instruction, DC 12-4
Digitizing with the Plotter 12-4
Manual Method 12-4
Example — Digitizing Using the Manual Method 12-5
Monitoring the Status Byte 12-5
Example — Digitizing by Monitoring the Status Byte 12-6
Example — Digitizing Many Points 12-6
HP-IB Interrupts and Polling 12-7

Chapter 13: Obtaining Information from the Plotter 13-1
What You'll Learn in This Chapter 13-1
HP-GL Instructions Covered 13-1
Other HP-GL Output Instructions 13-1
Hints for Obtaining Plotter Output Responses 13-2

ManualsPlus.com

Table of Contents (Continued)

Chapter 13: Obtaining Information from the Plotter (Continued)
Notes for HP-IB Configurations 13-3
Notes for RS-232-C Configurations 13-4
The Input Mask Instruction, IM 13-4
The Output Actual Position and Pen Status Instruction, OA 13-7
The Output Commanded Position and Pen Status Instruction, OC 13-8
The Output Error Instruction, OE 13-8
The Output Factors Instruction, OF 13-9
The Output Identification Instruction, OI 13-10
The Output Options Instruction, OO 13-10
The Output Status Instruction, OS 13-11
The Output Carousel Type Instruction, OT 13-12
Summary of Output Response Types 13-13

Chapter 14: Device-Control Instructions 14-1
What You'll Learn in This Chapter 14-1
Device-Control Instructions Covered 14-1
Terms You Should Understand 14-2
What Is a Device-Control Instruction? 14-2
How and When Do You Send a Device-Control Instruction to the Plotter? 14-3
How Do You Receive Information from the Plotter? 14-3
Syntax for Device-Control Instructions 14-4
The Allocate Configurable Memory Instruction, ESC . T 14-5
Hints for Using the ESC . T Instruction to Allocate Memory 14-8
How the Plotter Allocates Memory When More Than 12 800 Bytes Have Been Specified in ESC . T 14-10
The Output Buffer Size When Empty Instruction, ESC . L 14-11
The Output Configurable Memory Size Instruction, ESC . S 14-11
The Set Plotter Configuration Instruction, ESC . @ 14-12
Example — Increasing the Size of the Logical I/O Buffer and Setting the Configuration Bits 14-15
The Output Buffer Space Instruction, ESC . B 14-15
The Output Extended Error Instruction, ESC . E 14-16
Using the ESC . @ and ESC . E Instructions for Block I/O Error Checking (RS-232-C Only) 14-17
The Output Extended Status Instruction, ESC . O 14-19
The Set Monitor Mode Instruction, ESC . Q 14-22
Notes for HP-IB Configurations 14-23
Notes for RS-232-C Configurations 14-23

ManualsPlus.com

Table of Contents (Continued)

Chapter 14: Device-Control Instructions (Continued)
Data Flow in the Monitor Modes
(Any Interface Configuration) 14-24
The Abort Graphics Instruction, ESC . K 14-25
The Reset Instruction, ESC . R 14-26
The Output Identification Instruction, ESC . A 14-27
The End Flush Mode Instruction, ESC . U 14-27

Chapter 15: HP-IB Interfacing 15-1
What You'll Learn in This Chapter 15-1
What Is the HP-IB? 15-1
Addressing the Plotter 15-2
Addressable Mode 15-2
Listen-Only Mode 15-2
Notes on Addressing Protocol 15-3
Sending and Receiving Data 15-3
Computer-to-Plotter Examples 15-3
Plotter-to-Computer Examples 15-6
An HP-IB Overview 15-8
HP-IB Lines 15-8
HP-IB Operations 15-10
Addressing Sequences 15-11
Interface Functions 15-13
Bus Capabilities 15-14
Serial and Parallel Polling 15-17
The Serial Poll 15-17
The Parallel Poll 15-17

Chapter 16: RS-232-C/CCITT V.24 Interfacing 16-1
What You'll Learn in This Chapter 16-1
Device-Control Instructions Covered 16-1
Terms You Should Understand 16-2
Introduction 16-2
Plotter Configurations 16-3
Eavesdrop Configuration 16-3
Stand-Alone Configuration 16-4
Modes of Operation 16-5
Remote Mode 16-5
Local Mode 16-5
Standby Mode 16-6
A Description of Plotter Modes in the Eavesdrop and
Stand-Alone Configurations 16-6
Remote Eavesdrop Mode, Programmed-Off 16-6

ManualsPlus.com

Table of Contents (Continued)

Chapter 16: RS-232-C/CCITT V.24 Interfacing (Continued)
Remote Eavesdrop Mode, Programmed-On 16-7
Local Eavesdrop Mode, Programmed-Off 16-8
Local Eavesdrop Mode, Programmed-On 16-8
Remote Stand-Alone Mode 16-9
Local Stand-Alone Mode 16-9
Automatic Disconnection Modes 16-10
Switched/Datex-Line Disconnection Mode 16-10
Leased-Line Disconnection Mode 16-10
Direct/Modem Mode 16-10
Mechanical Interface and Connector Pin Allocations 16-11
RS-232-C/CCITT V.24 Interface Implementation 16-11
RS-422-A Interface Implementation 16-11
Connector Pin Allocations 16-11
Transmission Errors 16-18
Setting Up the Plotter: a Checklist 16-18
Determine System Configuration 16-18
Select Data Source 16-19
Check Hardware Connections 16-19
Set Baud Rate and Stop Bits 16-19
Set Parity and Data Bits 16-19
Set Direct or Modem Mode 16-20
Set Echo On or Off 16-20
Select a Handshake Method 16-20
Understanding HP 7550 Predefined Handshake Methods 16-20
Choosing a Handshake Method 16-21
Xon-Xoff Handshake 16-22
Enquire/Acknowledge Handshake 16-23
Hardwire Handshake 16-24
Software Checking Handshake 16-24
Selecting a Predefined Handshake Method from a Program ... 16-26
The Set Handshake Mode Instruction, ESC . P 16-27
Controlling Plotter Output 16-28
The Set Output Mode Instruction, ESC . M 16-28
The Set Extended Output and Handshake Mode Instruction, ESC . N 16-31
Tailoring Your Own Handshake 16-31
Quick Reference of Handshake Types 16-34
Device-Control Instructions and Values Used to Establish Handshake Methods 16-34
A Summary of RS-232-C Device-Control Instructions 16-35
The Set Handshake Mode 1 Instruction, ESC . H 16-35

ManualsPlus.com

Table of Contents (Continued)

Chapter 16: RS-232-C/CCITT V.24 Interfacing (Continued)
Example — Using ESC . H with an Enquire/Acknowledge Handshake 16-37
The Set Handshake Mode 2 Instruction, ESC . I 16-37
Example — Using ESC . I with a Hardwire Handshake 16-38
Example — Using ESC . I with an Enquire/Acknowledge Handshake 16-39
Example — Using ESC . I with an Xon-Xoff Handshake 16-39
The Abort Device Control Instruction, ESC . J 16-39
The Plotter-On Instruction, ESC . Y or ESC . (..... 16-39
The Plotter-Off Instruction, ESC . Z or ESC .) 16-40

Appendix A: Reference Material A-1
Binary Coding and Conversions A-1
Binary-Decimal Conversions A-1
ASCII Character Codes A-2
Character Sets and ASCII Codes A-2
Default Conditions Established by the DF Instruction A-11
Default Conditions Established by the IN Instruction A-13
IN Instruction with the −1 Parameter A-14
Default P1 and P2 Coordinates A-14
The No Operation (NOP) Instructions A-14
Scaling without Using the SC Instruction A-15
The Scale Instruction, SC† A-18

Appendix B: Error Messages B-1

Appendix C: Instruction Summary C-1
HP-GL Instructions C-1
Device-Control Instructions C-22

Subject Index SI-1

†This instruction applies to plotters whose serial number prefix is 2520A or smaller.

ManualsPlus.com

ManualsPlus.com

Chapter 1
Introduction to Programming the Plotter

What You'll Learn in This Chapter

This chapter will help you become acquainted with your plotter and how you can make it work for you. You will be introduced to HP-GL (Hewlett-Packard Graphics Language) and device-control instructions, learning how they are presented in this manual. Finally, you will see HP-GL in action with a programming example that uses HP-GL to plot a pie chart.

What Can Your HP 7550 Plotter Do for You?

Your HP 7550 plotter was designed to provide you with the flexibility, speed, and intelligence necessary to produce almost any plot that you require. You can use eight pens at any given time, but you have a choice of several pen colors and pen widths, depending on what type of pen you are using. You can use standard fiber-tip pens (for paper), transparency fiber-tip pens (for overhead transparency film), roller-ball pens, and drafting pens. You can also use a variety of plotting media: chart paper, vellum, and double-matte polyester film. The Operation and Interconnection Manual suggests which pens produce the best results on each type of plotting medium. The plotter accepts four standard media sizes:

ISO*	A4	(210 × 297 mm)	ANSI*	A	(8.5 × 11 in.)
ISO	A3	(297 × 420 mm)	ANSI	B	(11 × 17 in.)

Many of the plotter's features can be accessed from the front panel. This means you don't have to learn to program the plotter to use many of its features, even if you are using a purchased graphics software package. For full details of how to use the front panel, refer to the Operation and Interconnection Manual. Here are some examples of what you can do from the front panel. First, you can cause paper to be loaded automatically from the media loading tray, plus tell the plotter to replot a drawing a

*International Standards Organization (ISO)
American National Standards Institute (ANSI)

ManualsPlus.com

given number of times. Thus, if you need multiple copies of one plot, you can tell the plotter and then walk away until it has finished all of the copies. You can also select pens, increase their force to a maximum of 66 grams, and increase their speed to a maximum of 80 cm/s (31.5 in./s).

Finally, you can set up your interface from the front panel. If you are using an HP-IB (IEEE 488-1978) interface, you can change the HP-IB address. If you are using an RS-232-C/CCITT V.24* interface, you can set up standard handshakes and communication rates. (The Operation and Interconnection Manual tells you how. However, if you find that you need to set conditions not available from the front panel, you might need to set up the handshake using device-control instructions in a program. These instructions are described in this manual in Chapter 16.)

You can also program the plotter to do everything you can do from the front panel, plus a variety of sophisticated tasks. For example, you can issue special graphics instructions known as HP-GL (Hewlett-Packard Graphics Language) to have the plotter do automatic area-fill for any polygon shape, and label with any of 20 international character sets in two fonts. This manual provides full details on the capabilities provided by the HP-GL instruction set. A glance at the following table will give you an idea of the full range of capabilities provided by HP-GL. The chapter references are provided so that you can find an instruction quickly if you are already familiar with HP-GL.

Plotter Instruction Set

Instruction	Description	Chapter
AA	Arc absolute	6
AP	Automatic pen operations	10
AR	Arc relative	6
AS	Acceleration select	10
BF	Buffer plot	10
BL	Buffer label	7
CA	Designate alternate character set	11
CC	Character chord angle	11
CI	Circle	6
CM	Character selection mode	11
CP	Character plot	7
CS	Designate standard character set	11
CT	Chord tolerance	6
CV	Curved line generator	10
DC	Digitize clear	12

*All information in this manual applies equally to RS-232-C and CCITT V.24 interfaces, except where noted. For purposes of simplicity, both are referred to as RS-232-C.

ManualsPlus.com

Plotter Instruction Set (Continued)

Instruction	Description	Chapter
DF	Default	3
DI	Absolute direction	7
DL	Define downloadable character	11
DP	Digitize point	12
DR	Relative direction	7
DS	Designate character set into slot	11
DT	Define label terminator	7
EA	Edge rectangle absolute	6
EP	Edge polygon	6
ER	Edge rectangle relative	6
ES	Extra space	7
EW	Edge wedge	6
FP	Fill polygon	6
FS	Force select	10
FT	Fill type	6
GC	Group count	10
GM	Graphics memory	3
IM	Input mask	13
IN	Initialize	3
IP	Input P1 and P2	3
IV	Invoke character slot	11
IW	Input window	9
KY	Define key	10
LB	Label	7
LO	Label origin	7
LT	Line type	5
NR	Not ready (unload page)	10
OA	Output actual position and pen status	13
OC	Output commanded position and pen status	13
OD	Output digitized point and pen status	12
OE	Output error	13
OF	Output factors	13
OG	Output group count	10
OH	Output hard-clip limits	9
OI	Output identification	13
OK	Output key	10
OL	Output label length	7
OO	Output options	13

ManualsPlus.com

Plotter Instruction Set (Continued)

Instruction	Description	Chapter
OP	Output P1 and P2	9
OS	Output status	13
OT	Output carousel type	13
OW	Output window	9
PA	Plot absolute	4
PB	Print buffered label	7
PD	Pen down	4
PG	Page feed	10
PM	Polygon mode	6
PR	Plot relative	4
PT	Pen thickness	6
PU	Pen up	4
RA	Fill rectangle absolute	6
RO	Rotate coordinate system	9
RP	Replot	10
RR	Fill rectangle relative	6
SA	Select alternate character set	11
SC	Scale	3
SI	Absolute character size	7
SL	Character slant	7
SM	Symbol mode	5
SP	Select pen	4
SR	Relative character size	7
SS	Select standard character set	11
TL	Tick length	5
UC	User-defined character	11
UF	User-defined fill type	6
VS	Velocity select	10
WD	Write to display	10
WG	Fill wedge	6
XT	X-tick	5
YT	Y-tick	5

If you have advanced applications, you should also note that the plotter has a large 12 800-byte buffer, or "graphics memory." You can use the graphics memory for replot functions, as well as for polygon, character-designing, and input/output (I/O) functions. And, most importantly, you can change the amount of memory devoted to each function so that unused memory isn't wasted. Thus, if you are only using one function, you can allocate (assign) all of the plotter's memory to that one function. Refer to Chapter 14 for information on how to allocate graphics memory.

ManualsPlus.com

Programming Languages — Which Should You Use?

In order to write graphics programs for the plotter, you will need to learn the programming languages that the computer and the plotter understand. This manual assumes that you already have some general experience programming with your computer's language. You can use any computer that can output in ASCII* and any language that outputs literal strings.

To help you understand how your computer communicates with the plotter, some programming languages are described next. Begin with the plotter's HP-GL instructions. Then read about your computer's language to see how it communicates with the plotter. Three common computer languages are described: BASIC, FORTRAN, and Pascal. If you are programming in another language, your computer documentation should tell you how to output literal strings to a peripheral (in this case, literal strings of HP-GL to the plotter).

HP-GL and Device-Control Instructions

HP-GL

HP-GL instructions are codes that access the plotter's graphics functions. Each instruction consists of a two-letter mnemonic designed to remind you of its function. For example, SP is the select pen instruction, and PD is the pen down instruction. Often these two-letter mnemonics are followed by numerical parameters that tell the plotter how to execute (complete) the instruction. A typical HP-GL instruction looks like this:

```
PD 1500 , 1500 ;
```

To program the plotter, you must *insert the HP-GL instructions with their parameters in an appropriate output statement from your computer's language.* This is referred to as "sending" an HP-GL instruction to the plotter. You'll learn more about this in the section How to Use the Examples in this Manual, later in this chapter. When you send HP-GL instructions to the plotter, they enter the plotter's internal buffer; the plotter executes them on a first-in/first-out basis.

Device-Control Instructions (RS-232-C Only)

Device-control instructions control operations such as data formatting, input/output, handshaking, and buffer allocation. A device-control

*American Standard Code for Information Exchange

ManualsPlus.com

instruction is a three-character sequence which may or may not have parameters. A typical device-control instruction looks like this:

ESC . L

Device-control instructions are sent to the plotter as literal strings in a manner similar to sending HP-GL instructions. *Unlike HP-GL, however, device-control instructions do not enter the plotter's buffer and are executed immediately upon receipt.* Chapters 14 and 16 contain full details on these instructions.

BASIC

BASIC (Beginner's All-purpose Symbolic Instruction Code) is a common programming language on many computers. It uses statements that resemble English to perform many complex operations. Some graphics statements may be included in your implementation of BASIC for the CRT, but not necessarily for the plotter.

If you have an HP Series 80 or HP Series 200 computer, graphics statements (sometimes called AGL — A Graphics Language) for the plotter are available as an extension to BASIC. Like other BASIC statements, these statements resemble English, but they translate their functions into HP-GL so that the plotter can understand them. Refer to this chapter's section titled HP Series 80 and Series 200 Computers — A Note to Users.

If you do not have an HP Series 80 or HP Series 200 computer, your computer's BASIC graphics statements aren't translated into HP-GL for the plotter. Therefore, you will need to program the plotter directly by using BASIC output commands to send HP-GL strings.

FORTRAN and Pascal

FORTRAN (FORmula TRANslator) and Pascal are high-level programming languages. They generally do not include graphics statements that translate functions into HP-GL. Therefore you must program the plotter by sending strings of HP-GL instructions in a WRITE or WRITELN statement. (Refer to Examples Presented as HP-GL Strings in this chapter.)

How to Use the Examples in This Manual

The examples in this manual are designed primarily to show the use of the instruction with which they appear. If you are new to programming, try entering and running some examples on your computer. You might then wish to change some parameters in an instruction and rerun the

ManualsPlus.com

program. The examples are presented in two ways, either as complete programs or as listings of only the pertinent HP-GL strings. These two types of examples are described in the following paragraphs.

Examples Presented as Complete Programs

Some examples are presented as complete programs, written in a version of Microsoft® BASIC for MS™-DOS operating systems. This BASIC is used by the HP Touchscreen and HP Touchscreen Max, the IBM PC, and several other popular personal computers. (If you are using an RS-232-C interface, be sure you have established the proper handshaking protocol before running these programs. Refer to the Operating Manual and Chapter 16 of this manual.)

You will always need a configuration statement at the beginning of your program. It will be specific to your computer and the type of interface (RS-232-C or HP-IB) you are using.

If you are using Microsoft® BASIC, you can enter and run these programs as shown; just be sure to have the proper configuration statement for your computer in line 10.

If you are not using Microsoft® BASIC, you will need to insert the proper configuration statement in line 10 *and* may need to change the PRINT #1 or INPUT #1 statements as well.

The next two sections in this chapter, Examples of Opening Configuration Statements and Examples of Other BASIC Statements, will show you how to make these changes on some computers. Or, refer to Chapter 6, Plotter Interconnection, in the Operation and Interconnection Manual and use the interconnection program as a guide.

If your computer is not listed, refer to your computer documentation to learn how to make your computer and plotter communicate.

Examples of Opening Configuration Statements

Following are common forms of a configuration statement that can be used in line 10 of the BASIC programming examples. Refer to your computer documentation for a complete discussion of the parameters.

HP Touchscreen (150) RS-232-C and HP-IB interfaces

```
10 OPEN "O",1,"PLT"          (Series 100/BASIC)

10 OPEN "LPT3:" FOR OUTPUT AS #1
```

IBM PC/PC-XT/AT RS-232-C interface
AT&T PC 6300

```
10 OPEN "COM1:9600,N,8,1,RS,CS65535,DS,CD" AS #1
```

ManualsPlus.com

Apple IIc RS-232-C and IEEE-488 interfaces

```
10 PRINT CHR$(4); "PR#n"
```
(for slot number n)

Apple IIe/II Plus RS-232-C and IEEE-488 interfaces

```
10 PR#n
```
(for slot number n)

Examples of Other BASIC Statements

You might need to change other BASIC statements if the BASIC on your computer is different from the MS™-DOS version used here, or if you are programming in another language. Your computer's documentation should tell you how to do this. Here are the statements that you will likely need to change:

PRINT #1,	This statement sends the HP-GL instructions, via an output file, to the plotter. Your computer might use a statement such as WRITE, WRITELN, OUTPUT, LPRINT, or simply PRINT. Here **"1"** corresponds to the file number in the "OPEN" statement.
INPUT #1	This statement causes information from the plotter to be read by the computer. Your computer might use a statement such as READ, READLN, or ENTER. Here **"1"** corresponds to the file number in the OPEN statement.
FOR . . . NEXT **X = 3.14**	FOR . . . NEXT are loop statements. X = 3.14 is a variable assignment. Change these statements to whatever is comparable in your language.

Examples Presented as HP-GL Strings

Since input/output instructions (e.g., PRINT # and INPUT #) do vary so much among different computers, some examples present only the pertinent HP-GL strings (the two-letter mnemonics and applicable parameters). They are enclosed in quotation marks since most languages use quotation marks as string delimiters. *The plotter does not require the quotation marks; use whatever your* **computer** *requires.* Add the statements required by your system to send the string of instructions. For example, suppose this string of HP-GL instructions is printed in the manual:

```
"SP1;"
```

First use whatever opening statements are required to define the computer's output port and establish the plotter as the recipient of an

ManualsPlus.com

output string. Then, send the string within a statement similar to one shown here, depending on your computer and language (only a few computers are listed here to give you an idea of the variety of statements available):

HP Series 200, BASIC	`OUTPUT 705; "SP1;"`
	or
	`PRINT "SP1;"`

HP Touchscreen (150) Computer IBM PC/PC-XT/AT	`PRINT #1, "SP1;"`
HP 9000, Pascal	`WRITELN('SP1;')`
HP 3000, FORTRAN	`WRITE (6,10)` `10 FORMAT(X,4HSP1;)`

HP Series 80 and HP Series 200 Computers — A Note to Users

If you use BASIC on an HP Series 80* or HP Series 200 computer (formerly models HP 9816, HP 9826 and HP 9836), you can use high-level BASIC graphics statements to program your plotter. These computers actually encode, or translate, the BASIC graphics statements into HP-GL instructions before sending them on to the plotter. For this reason, you might find it easier to use these graphics statements for much of your plotting.

For example, compare the BASIC graphics statements CLIP and GRID with their equivalent HP-GL instructions: PA, PD, TL, XT, and YT. In either case, the plotter draws grid lines every 10 units within a rectangle, as shown. Yet, the CLIP and GRID statements are easier to enter in the computer.

*Series 80 computers must contain a plotter ROM: Part No. 00085-15002 for the HP-85, and Part No. 00087-15002 for the HP-86 and HP-87.

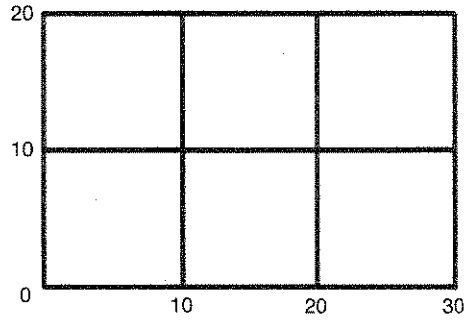
ManualsPlus.com

BASIC Graphics

```
CLIP 0,30,0,20
GRID 10,10,0,0
```

HP-GL

```
TL 100; PA 30,0; PD; XT;
PA 20,0; XT; PA 10,0; XT;
PA 0,0; PA 0,10; YT; PA 0,20; YT;
```



NOTE: This example is not a complete program. It requires additional scaling statements to establish the plotting area and units. These statements were omitted here to help clarify the direct comparison between the BASIC graphics statements and their equivalent instructions. ■

Understanding Manual Conventions

Before reading the rest of this manual, you should understand the meaning of type styles, symbols, and number representation used in the text. Words typed in **SMALL BOLDFACE TYPE** are either keys, switches, or words actually found on the plotter or computer. **BOLD CONDENSED TYPE** denotes a single ASCII character which should be sent to the plotter. Numbers are typed using SI (International System of Units) standards; numbers with more than four digits are placed in groups of three, separated by a space, counting both to the left and right of the decimal point (54 321.123 45).

The symbols that represent the syntax requirements of HP-GL and device-control instructions are discussed in Chapters 3 and 14, respectively.

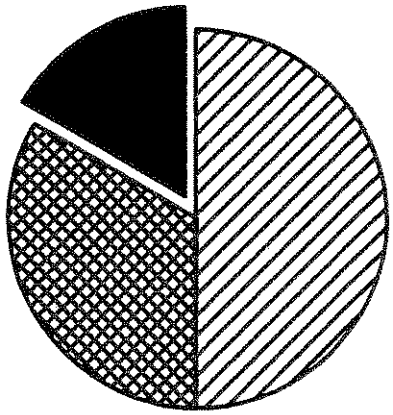
Developing a Plot

You are probably anxious to see HP-GL in action. Following is a very simple program for plotting a pie chart with three sectors. This program is presented here because there are only two BASIC statements that you might have to change for your system — the configuration statement for your computer in line 10 and PRINT #.

ManualsPlus.com

For now, key in the following program. If you are using the RS-232-C interface, make sure your handshaking protocol is established. (Refer to the Operation and Interconnection Manual or Chapter 16 of this manual.) Then make sure your plotter and computer are properly connected.

Finally, run the program to see the plotted results.



```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP2;PA7000,2000;"
30  PRINT #1, "FT3,75,45;"
40  PRINT #1, "WG-1000,90,180;"
45  PRINT #1, "EP;"
50  PRINT #1, "SP3;PR-60,110;FT1,0,0;"
60  PRINT #1, "WG-1000,270,60;"
65  PRINT #1, "EP;"
70  PRINT #1, "SP4;PA7000,2000;FT4,60,45;"
80  PRINT #1, "WG-1000,330,120;"
85  PRINT #1, "EP;"
90  PRINT #1, "SP0;"
100 END
```

The program is explained fully under The Wedge Instructions, WG and EW, in Chapter 6. Here we'll just discuss the concept of organizing a program.

Lines 10, 20, 90, and 100 show how you might typically begin and end every program. Line 10 is a computer-dependent statement that identifies the plotter as the recipient of the strings of HP-GL instructions, which are sent in PRINT # statements.

Line 20 issues the following instructions: initialize, IN; select pen, SP; and plot absolute, PA. Brief descriptions of each instruction follow. For more details, refer to Chapters 3 and 4.

ManualsPlus.com

- IN resets the plotter to the conditions that exist when the plotter is first turned on. In other words, any conditions that might have been set in the plotter by a previous program are cleared (defaulted) before your plot is drawn. In addition, most conditions that might have been set in the plotter from the front panel are cleared. All examples in this manual begin with IN to ensure that they are plotted as shown. However, in your own programs, you might find it more convenient to use the default instruction, DF, instead of the IN instruction. DF is a modified version of IN; while DF clears most HP-GL conditions, it does not affect most front-panel settings. This is particularly useful, for example, if you want to run a program several times and vary the pen speed or the settings of the scaling points P1 and P2. You can make the settings from the front panel and then run the program. (If you used IN instead of DF, the front-panel settings would be cleared by the IN instruction, and nothing would change in the way the program was running.)
- SP causes the pen holder to take the specified pen from the carousel. If you do not issue an SP instruction and there is not a pen currently in the pen holder, the pen holder will go through the motions of drawing the plot, but the page will remain blank.
- The PA instruction establishes the beginning pen position (in this case, it is the center of the pie chart).

Lines 30 through 85 are the body of the program; their organization and content depend on the purpose of the program.

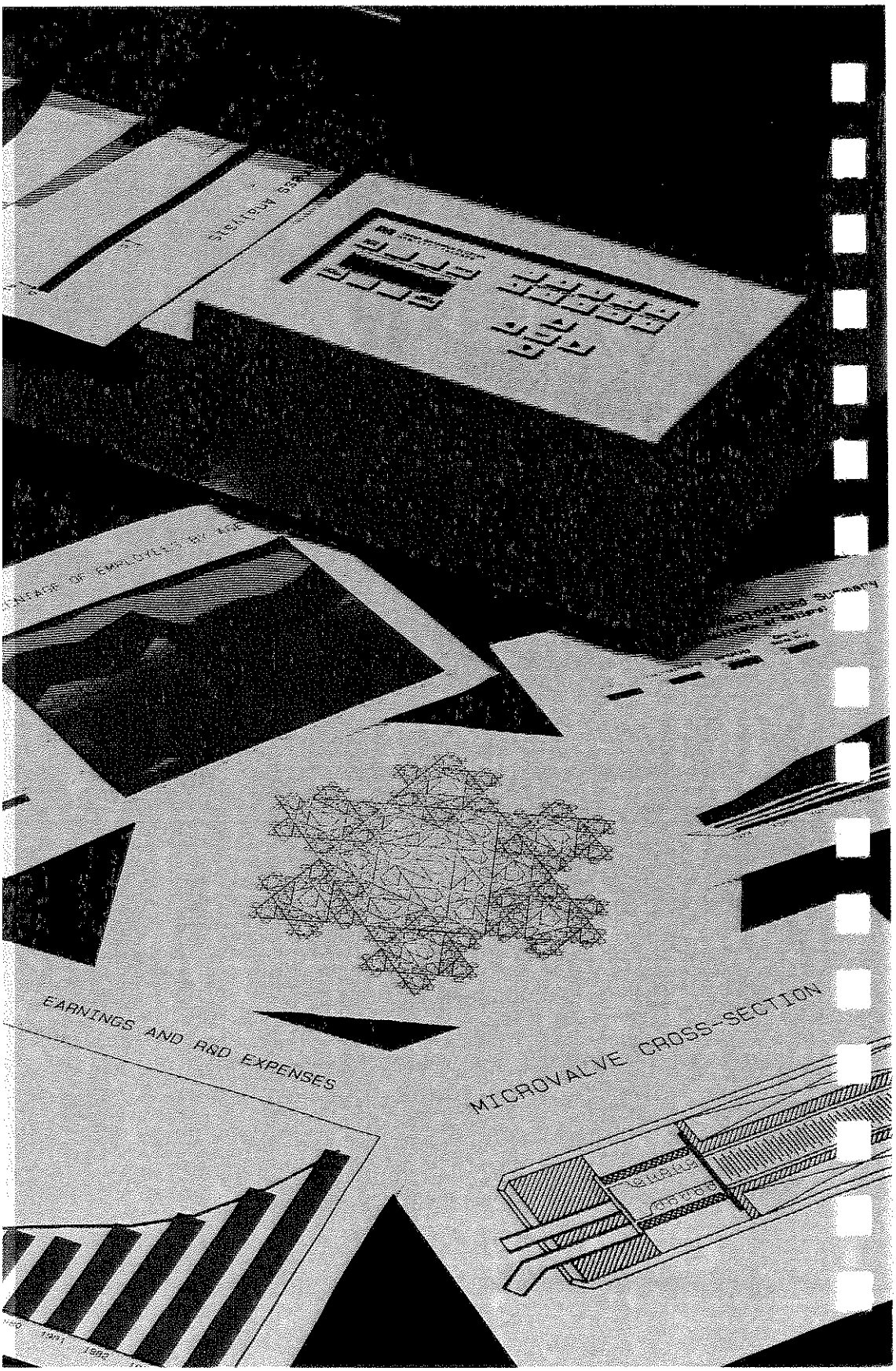
Line 90 uses the SP instruction to return the pen to the carousel. This is a good idea to prevent the pen from drying out, and to signal the completion of the plot. You might also wish to include the not-ready instruction, NR, to unload the plot (refer to Chapter 10). This prevents another program from drawing over your plot before you have a chance to remove it from the plotter.

Line 100 is a BASIC statement that ends the execution of the program.

ManualsPlus.com

Notes

ManualsPlus.com



ManualsPlus.com

Chapter 2
Fundamental Plotting Concepts

What You'll Learn in This Chapter

In this chapter, you will gain the foundation necessary to understand hardcopy plotting concepts. For example, in order to use the plotting instructions, you need to be able to specify a location on the paper. This chapter explains the Cartesian coordinate system which the plotter uses for locating points, and discusses the units of measure that the plotter understands. You will also learn about graphics limits, pen status, and absolute and relative coordinates. This chapter will be useful to you if you have never done any graphics programming or if you have written programs for a graphics terminal but not for a plotter.

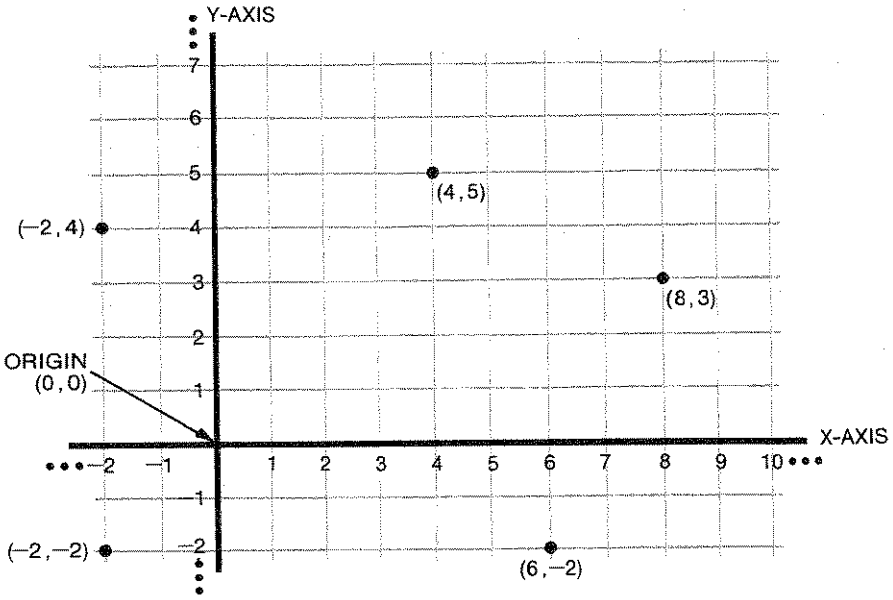
Introduction to the Cartesian Coordinate System

The plotting area is that portion of the paper in which the plotter's pen can draw. This plotting area is divided into a grid, as shown in the illustration on the next page. Each grid line represents a unit of measurement in one of two directions — horizontal or vertical. The horizontal direction is known as the X-axis, whereas the vertical direction is known as the Y-axis. The intersection of these two axes provides a convenient reference point known as the origin. To locate a position (or point) on the plotting area, you simply tell the plotter how many X-axis units and how many Y-axis units to move away from the origin. These are known as the X-coordinate and the Y-coordinate; together, they are an X,Y coordinate pair. To specify a position, you must always specify a complete X,Y coordinate pair, with the X-coordinate first and the Y-coordinate second. This system of locating points is known as a two-dimensional Cartesian coordinate system. (Cartesian comes from the name of the French mathematician René Descartes, who invented the grid system.)

The origin is defined as the X,Y coordinate pair (or point) 0 , 0. If you move to the right of or above the origin, you specify a positive X-coordinate or Y-coordinate, respectively. Likewise, if you move to the left of or below the

ManualsPlus.com

origin, you specify a negative X-coordinate or Y-coordinate, respectively. Look at the illustration again to locate these points: 0 , 0; 4 , 5; −2 , 4; −2 , −2; 6 , −2; 8 , 3.



Cartesian Coordinate System

Plotting with the Coordinate System

HP-GL instructs the plotter what to do and where to do it. Thus, you must specify X,Y coordinates as the parameters in many HP-GL instructions so that the plotter will know where to plot. Of course, you can't specify exactly where an X,Y coordinate lies on the paper without some unit of measurement.

You can express coordinates with two types of units, called plotter units and user units. Plotter units are a constant, fixed size, and are the units that the plotter always uses internally. When the plotter is turned on, it assumes that your coordinates are expressed in plotter units. But you might find that some other unit makes more sense to you and your application. This is called a user unit because it is up to you, as the plotter's user, to define the unit. You can define different user units for each plot. If you define user units, the plotter automatically converts your coordinates to plotter units so that it can find the correct location on the paper. This conversion is internal to the plotter, so you can express your coordinates in user units without ever calculating their equivalent plotter units. Thus, user units are very convenient to use for plotting.

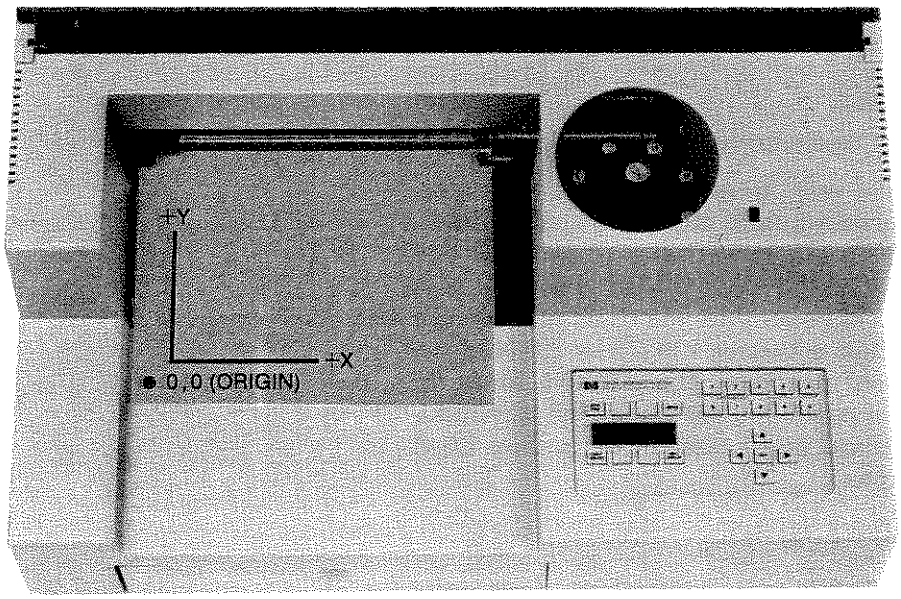
ManualsPlus.com

Both plotter units and user units are discussed in detail in the following paragraphs.

Plotter Units

The plotter assigns a measurement known as a plotter unit to each "grid line" discussed previously. A plotter unit is the smallest move the plotter can make (this is also known as the addressable resolution of the plotter). A plotter unit is equal to 0.025 mm (0.000 98 in.) in length; alternatively, 40 plotter units equal 1 mm, or 1016 plotter units equal 1 in. Because a plotter unit is the smallest move the plotter can make, it is interpreted as an integer.*

The plotter-unit coordinate system is oriented on the paper so that the origin (0 , 0) always lies inside the lower-left corner of the paper with the Y-axis extending upward along the short side of the paper, and the X-axis extending to the right along the long side of the paper.** The orientation on the paper is the same, regardless of what direction the paper is loaded in the plotter, as shown in the following illustrations.

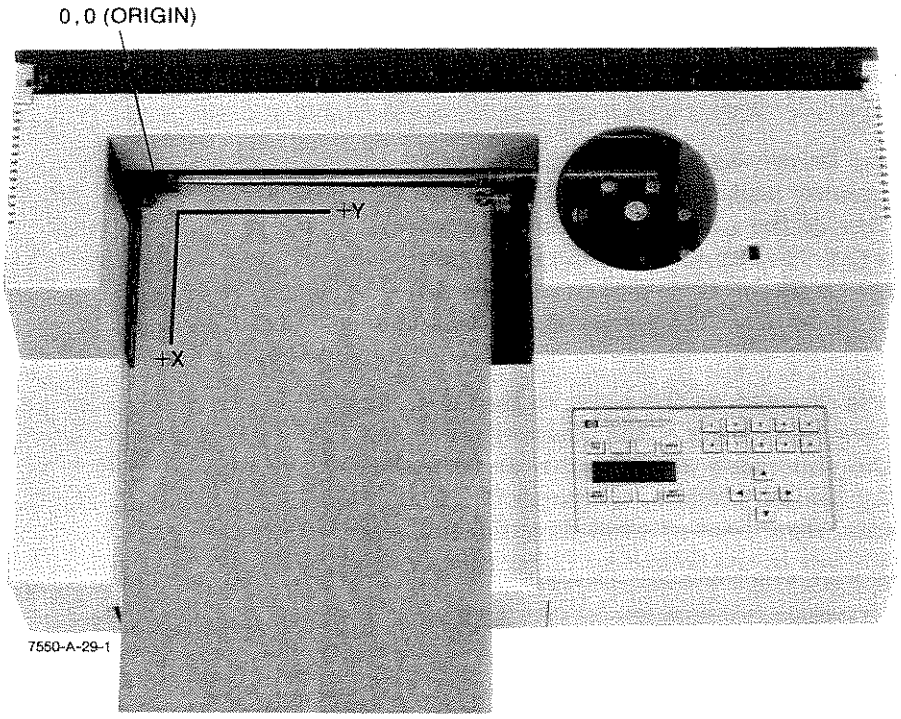


Default Orientation of Plotter Coordinate System (A4/A Paper)

*Some instructions allow you to express X,Y coordinates as decimal numbers (with fractions), even if you are using plotter units. In this case, the plotter will move using only the integer portion of the number. This process is described under Parameter Formats in the discussion on HP-GL Syntax in Chapter 3.

**You can change the orientation of this system with the rotate instruction, RO, or with the ROTATE function key on the front panel. The RO instruction is discussed in Chapter 9.

ManualsPlus.com



Default Orientation of Plotter Coordinate System (A3/B Paper)

You can see that the pen can plot only to positive X,Y coordinates. However, the permissable range for coordinates includes negative plotter units. In fact, the numerical range of plotter units that the plotter understands is -2^{23} to $+(2^{23}-1)$, or −8 388 608 to +8 388 607. Obviously, this is much larger than any sheet of paper that you could place on the plotter. (Imagine a piece of paper that measures 209 715 by 209 715 mm, or 8256 by 8256 in.!) For practical purposes, the effective range of plotter units is limited to the size of paper you are using. You can find a list of plotter-unit ranges for standard paper sizes in the section titled Hard-Clip Limits later in this chapter.

User Units

As mentioned before, you might find it more convenient to use units that fit your application. For example, you might want to make a bar chart with total sales in dollars along the Y-axis and number of months along the X-axis. It would be very cumbersome for you to convert your units into plotter units so that the plotter would understand the correct position on the paper. Fortunately, the plotter provides a way for you to specify your units so that the plotter will automatically convert them to plotter units.

ManualsPlus.com

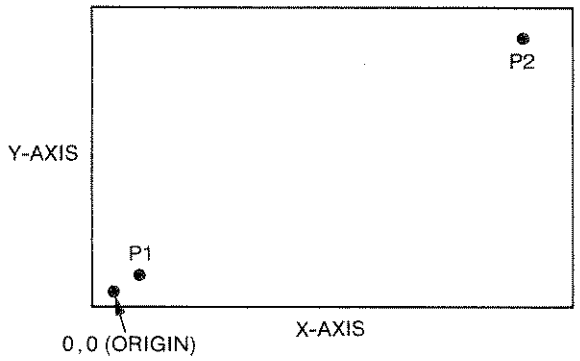
This process is known as scaling, because in effect, you are assigning a scale of user units to plotter units. For example, consider a map whose scale is 100 miles per inch. In this case, the miles are equivalent to user units and the inch is equivalent to plotter units. The following section, Scaling, describes how to scale plotter units into user units.

User units are useful for several reasons. As mentioned, they are more convenient because they are more meaningful to you and your application. User units can be months, years, dollars, francs, distances, temperatures, population, or whatever makes sense to you.

In addition, user units are very flexible. You can assign the origin (0,0) to be anywhere on the paper (not just the lower-left corner, as with plotter units). This means you can specify negative X,Y coordinates (to show, e.g., losses in a line chart). It also means you can specify coordinates with fractions (instead of only integers, as with plotter units). Therefore, you can plot your data accurately. Note that the maximum numerical range for user units is the same as for plotter units, i.e., -2^{23} to $+(2^{23}-1)$, or $-8\,388\,608$ to $+8\,388\,607$. (Hint: If you want to plot data with values larger than this range, reduce your data by an arithmetic process before sending it to the plotter. For example, divide all data by some factor of 10 so that all data points are within the acceptable range.)

Scaling

Before we can discuss the process of scaling, you should understand what is meant by the scaling points P1 and P2. The plotter sets these points at power-on so that P1 defines the lower-left corner of a rectangular area on the paper, and P2 defines the upper-right corner. At power-on, P1 and P2 always represent an absolute plotter-unit location on the paper with an orientation on the paper as shown in the next illustration. The actual locations of P1 and P2 depend on the size of paper loaded in the plotter, as listed in the table following the illustration.



Default Orientation of P1 and P2 (Any Paper Size)

ManualsPlus.com

Default Coordinate Values for P1 and P2

Paper Size	Default Scaling Points (in Plotter Units)	
	$P1_X$,$P1_Y$	$P2_X$,$P2_Y$
A4 (210 × 297 mm)	430,200	10430,7400
A3 (297 × 420 mm)	380,430	15580,10430
A (8.5 × 11 in.)	80,320	10080,7520
B (11 × 17 in.)	620,80	15820,10080

You can change the plotter-unit locations of P1 and P2 either from the front panel or with the input P1 and P2 instruction, IP. Use the IP instruction if you want to be sure that P1 and P2 are always in the same location whenever you plot the program (refer to Chapter 3). Use the front-panel function keys if you want to change P1 and P2 just for the current plot (refer to the Operation and Interconnection Manual).

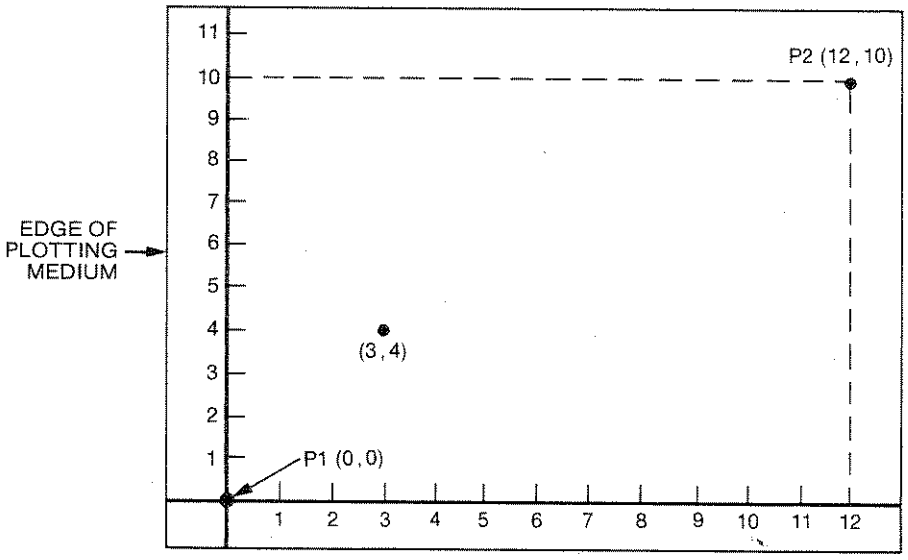
P1 and P2 are called "scaling points" because they take on the user-unit values that you specify in the scale instruction, SC. For details on the SC instruction, refer to Chapter 3. The scaling concepts discussed here only require that you know that the SC instruction specifies the minimum and maximum user-unit coordinates.

When you issue an SC instruction, P1 takes on the minimum X,Y coordinate values, and P2 takes on the maximum X,Y coordinate values.* This is sometimes also referred to as mapping; the coordinate values are "mapped onto" P1 and P2. The entire plotting area is then effectively divided into an imaginary grid based on the new user units. The actual size of the units depends on the locations of P1 and P2.

For example, suppose you want the Y-axis to be divided into 10 units representing thousands of dollars, and the X-axis to be divided into 12 units representing months, as shown on the next page. You would specify minimum coordinates of 0,0 and maximum coordinates of 12,10. P1 and P2 then take on these new user-unit values and the entire plotting area is divided into these units. Subsequent plotting instructions will now be interpreted using these units. This means that if you tell the plotter to move to the point 3,4, the plotter will move to the location equivalent to 3,4 user units, not 3,4 plotter units.

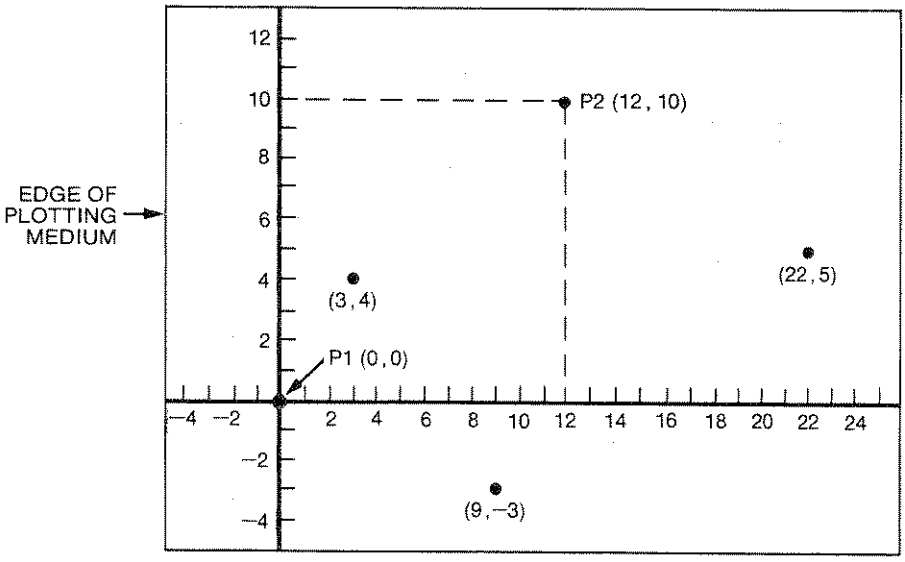
*Actually, it is possible to scale P1 and P2 so that P1 takes on the maximum coordinates and P2 takes on the minimum coordinates. However, this is not the usual method, so it is not discussed here. Refer to the explanations of the IP and SC instructions in Chapter 3, and to Techniques for Changing the Plotting Area in Chapter 9.

ManualsPlus.com



User-Unit Scaling with Default P1 and P2

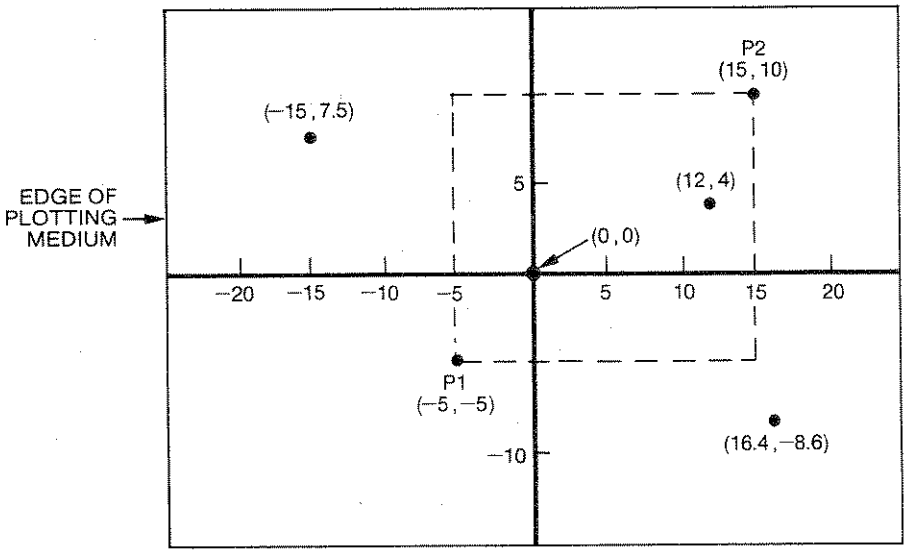
If the locations of P1 and P2 are moved, the sizes of the user units could change. The previous illustration showed P1 and P2 in their default locations. Following are the same minimum/maximum user units, only with different locations of P1 and P2. Note that the sizes of the user units are different along both the X- and Y-axes. Also, the range of units across the page has increased.



Same User-Unit Scaling with New P1 and P2

ManualsPlus.com

To further illustrate the flexibility of user-unit scaling, the following diagram shows P1 and P2 with the user-unit values of −5,−5 and 15,10 respectively. Remember that when user-unit scaling has been established, you can plot anywhere on the page using negative or positive values.



New P1 and P2, and User-Unit Scaling with Negative Values

An obvious benefit to user-unit scaling is that you can change the P1/P2 settings so that your plot occupies more or less space on the page. Also, you can scale onto default P1/P2 settings so that your plot will occupy the full area of any size of paper (A4/A or A3/B). Contrast this flexibility with plotting in plotter units: since plotter units are absolute with relation to a fixed origin, plots always occupy the same space on any size of paper.

You can see that there are many possibilities for changing your plotting area and your plot's proportions using P1/P2 and the SC instruction. Basic scaling information is presented in the explanations of the IP and SC instructions in Chapter 3. However, if you wish to learn about more advanced techniques, some are presented in Chapter 9. These techniques include placing more than one plot on a page, enlarging/reducing a plot, and plotting mirror images.

NOTE: P1 and P2 are mainly used in conjunction with the SC instruction to establish user-unit scaling. However, even when user-unit scaling has *not* been established, the plotter monitors the settings of P1 and P2 for interpretation of certain HP-GL instructions such as fill type, FT, line type, LT, relative character size, SR, and relative direction, DR. In addition, your computer or graphics software package might use P1

ManualsPlus.com

and P2 for other purposes, such as for establishing graphics limits. (Refer to the next section in this chapter, Graphics Limits.) ■

Graphics Limits

The plotter recognizes two basic types of graphics limits: the "hard-clip limits" and the "soft-clip limits." Their names imply their functions. First, "clip" indicates that some portion of the paper is being selected. Think of clipping an article out of a newspaper: by clipping away the edges, you select a smaller portion of the newspaper page.

"Hard-clip" refers to a physical boundary, the limits beyond which the pen physically cannot move. "Soft-clip," on the other hand, refers to a software-controlled boundary that temporarily limits pen movement within its borders. When the plotter is first turned on, the soft-clip limits are set equal to the hard-clip limits. Each of these concepts is discussed in more detail in the following paragraphs.

Hard-Clip Limits

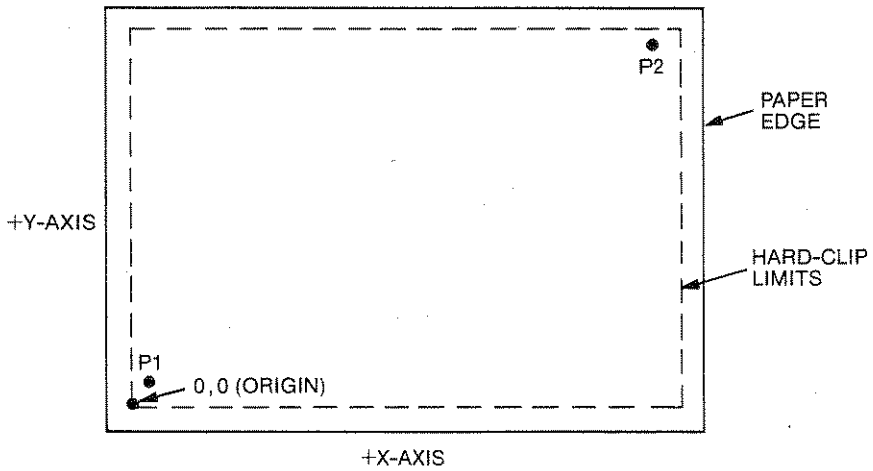
Naturally, you would not like your plotter to start plotting off of the edge of the paper. Therefore, the plotter automatically sets hard-clip limits when it senses the paper size. The hard-clip limits represent the physical boundary for pen movement. Thus, the limits are set a few millimetres within the edges of the paper to allow room for the pinch wheels to move without rolling over plotted lines and possibly smearing ink. The hard-clip limits correspond to the maximum plotting range. The maximum plotting ranges for standard paper sizes are listed in the following table.

Maximum Plotting Ranges

Paper Size	Maximum Plotting Range (in Plotter Units)	
	X-axis	Y-axis
A4 (210 × 297 mm)	0 - 10870 (271.75 mm/10.65 in.)	0 - 7600 (190.00 mm/7.45 in.)
A3 (297 × 420 mm)	0 - 15970 (399.25 mm/15.65 in.)	0 - 10870 (271.75 mm/10.65 in.)
A (8.5 × 11 in.)	0 - 10170 (254.25 mm/9.97 in.)	0 - 7840 (196.00 mm/7.68 in.)
B (11 × 17 in.)	0 - 16450 (411.25 mm/16.12 in.)	0 - 10170 (254.25 mm/9.97 in.)

ManualsPlus.com

The following illustration shows the relationships between the axis orientation, the P1/P2 scaling points, and the hard-clip limits. This is true for any size of paper when the plotter is first turned on.



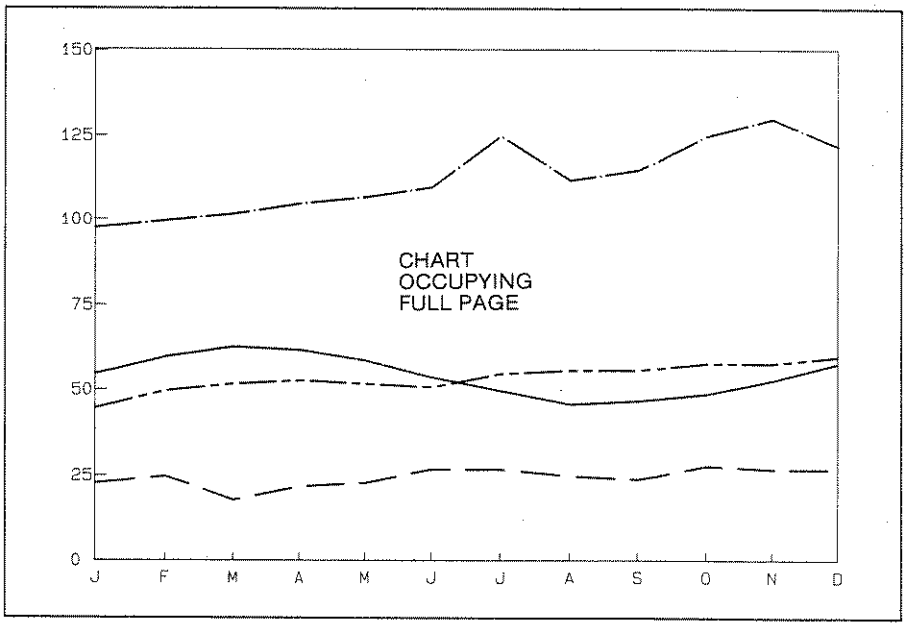
Hard-Clip Limits (Any Paper Size)

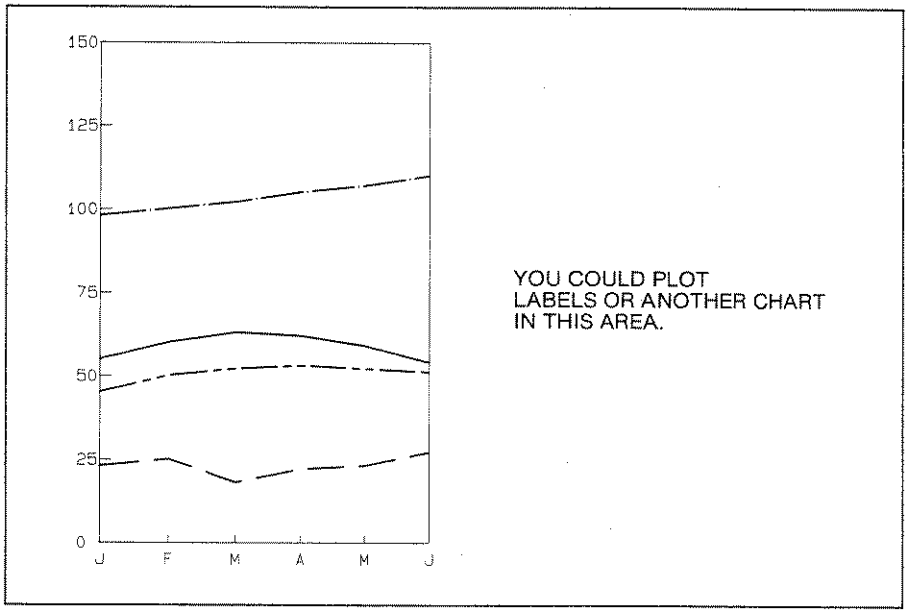
NOTE: You cannot change the hard-clip limits using HP-GL. However, some BASIC graphics statements implemented on HP computers do change the hard-clip limits. The PLOTTER IS statement, for example, usually reads the P1/P2 scaling points and sets the effective hard-clip limits to these points, rather than to the plotter's hard-clip limits. In this case, hard-clip limits relate to how the computer defines them, not to how the plotter defines them. For more information, refer to your computer's documentation on its BASIC graphics statements. ■

Soft-Clip Limits

As previously mentioned, soft-clip limits temporarily restrict pen movement to a specified area of the page. Usually you will use soft-clip limits to define a portion of the page when you want assurance that nothing will be drawn beyond that portion. For example, look at the following line chart. Suppose after plotting the full chart, you decide to make a new plot showing only the first 6 months of data so that you can use the space on the right for text. You could use the same program as for the full line chart, but add soft-clip limits to restrict plotting to the first half of the chart. Then you could add lines to the end of the program with new soft-clip limits and instructions for the labels.

ManualsPlus.com





SOFT-CLIP LIMITS CAUSE
ONLY THIS MUCH OF THE
PREVIOUS CHART TO BE PLOTTED.

ManualsPlus.com

When you look at the clipped area, it is almost like seeing part of the chart through a window. In fact, soft-clip limits are often referred to as windows. In HP-GL, use the input window instruction, IW, to define the lower-left and upper-right boundaries of a window. In other programming languages, refer to your documentation for the correct statements to use (in BASIC, try CLIP or WINDOW).

Soft-clip limits can be used in conjunction with user-unit scaling and the scaling points P1 and P2 to provide a great deal of flexibility with respect to where you plot data, how much of your data is plotted, and reducing/enlarging plots. The specifics are beyond the scope of this chapter. For more information, refer to Chapter 9.

Pen Movement

In addition to scaling and graphics limits, you should understand how the plotter finds the correct positions on the paper. Then you'll be ready to learn about the individual HP-GL instructions and put them to the best use for your application.

Current Pen Status and Pen Position

When you first turn on the plotter, it initializes itself and establishes the hard-clip limits and P1/P2 positions for the size of paper you have loaded. It also sets other pre-determined conditions, known as defaults. Among these defaults are the "pen status" and the "pen position." (You can find a full list of default conditions under The Default Instruction, DF, and The Initialize Instruction, IN, in Chapter 3.)

The pen status refers to whether or not a pen is loaded in the holder, and whether the pen is up or down. At power on, the plotter assumes there is no pen in the holder and the pen (holder) is up. This means no lines will be drawn if you issue a plotting instruction. Therefore, you must select a pen and lower it if you want something drawn on the paper. You can do this with front-panel function keys, or you can issue the select pen instruction, SP, and the pen down instruction, PD. The pen remains down for subsequent plotting instructions until you issue a pen up instruction, PU, or lift the pen using a front-panel function key. Also, the pen remains in the holder until you select a new pen or return the pen to the carousel.*

*The plotter automatically lifts the pen and returns it to the carousel if it remains inactive for an allotted time period. Although this causes the pen to be physically lifted and/or stored, the plotter does not sense it and does not update the current pen status. Refer to the Automatic Pen Operations Instruction, AP, in Chapter 10.

ManualsPlus.com

Every time you change pens or the up/down position, this pen status information is updated in the plotter. Most plotting instructions plot according to this concept of "current pen status." If you plot a program and portions of the plot aren't drawn, check your program to be sure you have positioned the pen down before the affected instruction(s). The descriptions of each plotting instruction tell you what happens to the current pen status (i.e., whether it is updated at the end of the instruction, or whether it is restored to the status that existed before the instruction was issued.)

The plotter keeps track of the current pen position in a similar manner. At power-on, the plotter sets the current pen position (in plotter units) to:

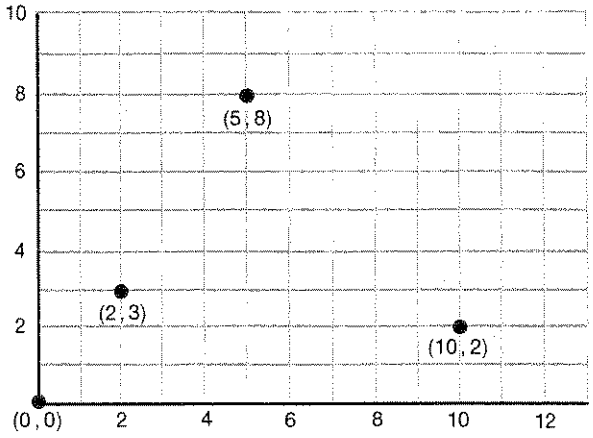
0,7600	A4-size paper
0,0	A3-size paper
0,7840	A-size paper
0,0	B-size paper

If you do not specify a different position, any plotting instruction will begin at one of the coordinates listed above, depending on the paper size. Whenever you issue a plot absolute or plot relative instruction, PA or PR, the pen begins at its current position and moves (according to the current pen status) to the new location specified by the PA or PR instruction. The plotter then updates the "current pen position" with the new location. Some plotting instructions do not cause the pen position to be updated. For example, no matter where the pen physically ends up after drawing a circle with the CI instruction, the plotter returns the pen to its exact position before the CI instruction was issued. In other words, the current pen position is restored after a CI instruction. As with the pen status, the descriptions of each instruction tell you whether the current pen position is updated or restored.

Absolute and Relative Movement

You know that the plotter interprets the pen position according to a coordinate system, and that you can specify coordinates using plotter units or user units. However, so far all discussions have assumed that you are specifying coordinates in *absolute* terms. Absolute means that the coordinates have an absolute, fixed position on the coordinate system with respect to the origin (0,0). In the following illustration, the coordinates 2,3; 5,8; and 10,2 are always in the same place with respect to the origin, no matter where the pen is when the coordinates are issued.

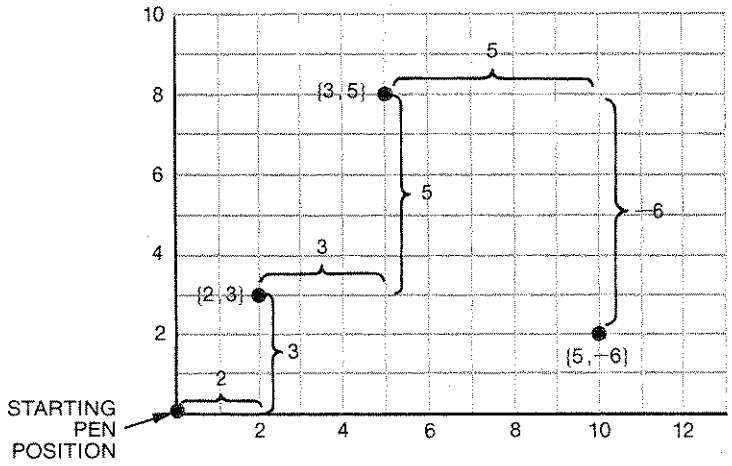
ManualsPlus.com



Absolute Coordinates

You can also specify coordinates whose location will be determined relative to the current pen position; these are known as relative coordinates. Relative coordinates are more accurately called increments, because their values represent the number of units for the pen to move from its current position. (As with absolute coordinates, the units can be user units or plotter units.)

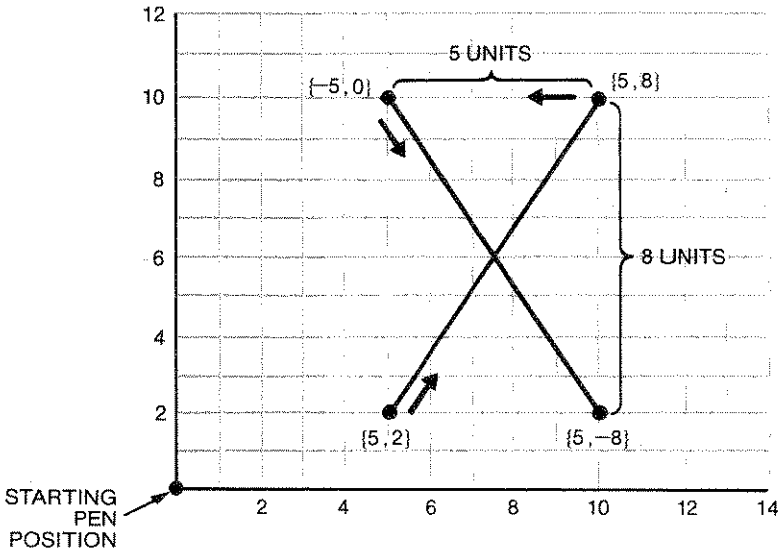
For example, assume that the pen is currently at the origin. To arrive at the lower-left point marked in the previous illustration, count 2 units to the right and 3 units above the current position. This is the relative position 2,3. Now count the X-units and the Y-units from this position to the upper point. You are now at the relative position 3,5. Continue for the lower-right point, arriving at the relative position 5,−6. The new coordinates are shown below.



Relative Coordinates

ManualsPlus.com

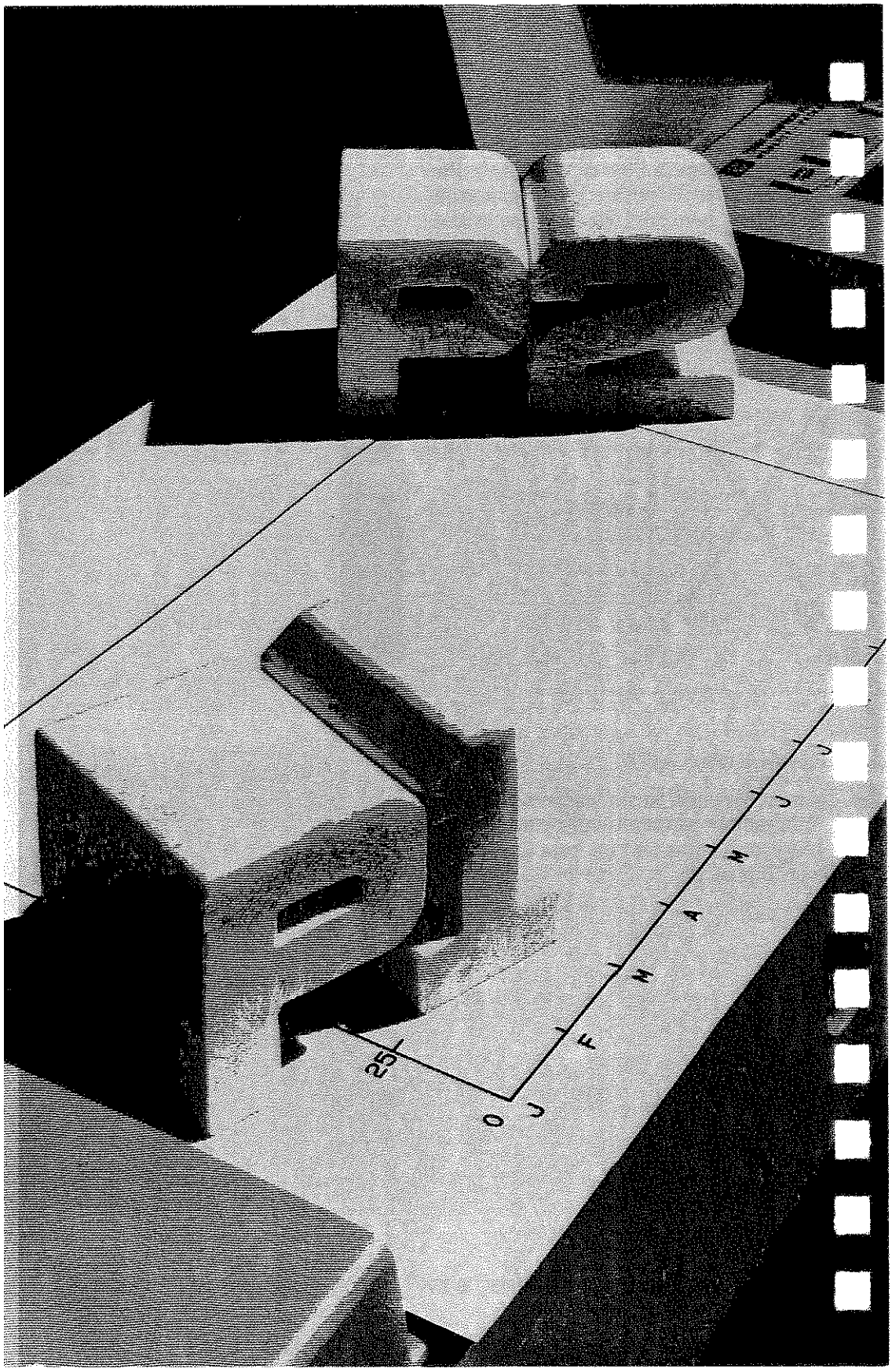
Only the plotting instructions that include "relative" in their title will interpret coordinates as relative increments. Relative plotting is particularly useful for shapes where you know the dimensions, but don't want to calculate the absolute coordinates for each endpoint. For example, suppose you want to plot the letter X with a width of 5 units and a height of 8 units, and the current pen position is at the origin. Here is the sequence of relative coordinates to draw the X: 5,2; 5,8; −5,0; 5,−8.



Relative Coordinates for Plotting "X"

NOTE: Relative coordinates (increments) add to the current pen position monitored by the plotter. The plotter automatically converts the new relative position to its absolute coordinates and updates the current monitored position in absolute coordinates. ■

ManualsPlus.com



ManualsPlus.com

Chapter 3
Preliminary Setup Using HP-GL

What You'll Learn in This Chapter

This chapter explains how to interpret HP-GL syntax, and introduces you to four HP-GL instructions. You will learn about initialization and default conditions, and you will learn how to scale the plotting area into user units that are meaningful to your application.

HP-GL Instructions Covered

DF The Default Instruction
IN The Initialize Instruction
IP The Input P1 and P2 Instruction
SC The Scale Instruction

Terms You Should Understand

String — a sequence of alphanumeric characters that are interpreted precisely as themselves, not as representing something else. HP-GL instructions and their parameters must be sent to the plotter as literal strings in a computer-dependent output statement (for example, PRINT, PRINT #, WRITE, or WRITELN).

Mnemonic — an abbreviation that is easy to remember. HP-GL instructions are two-letter mnemonics that represent the function of the instruction. For example, SP for select pen, and LT for line type.

Parameter — one or more characters following the HP-GL mnemonic that govern how the instruction is executed by the plotter. In the discussions of each instruction in this manual, the SYNTAX section specifies which parameters to include with the instruction and provides the range of values for the parameters.

Syntax — rules governing the structure of a language, such as HP-GL. In HP-GL, the syntax governs the sequence of mnemonics and parameters, the separators between mnemonics and parameters, and terminators at the end of a string.

ManualsPlus.com

Separator — a symbol that serves to separate each mnemonic and its parameters. The comma in this string is a separator: PA10 , 20. If there was no separator, the plotter would receive this string: PA1020. The plotter would interpret this string differently from the previous string.

Terminator — a symbol that signals the end of a string consisting of a mnemonic and its parameters.

Default — a known state. Default often refers to conditions that exist in the plotter when it is first turned on. Default also refers to the value of a parameter if you do not specify a different value.

Initialize — to set the plotter to all of the conditions that exist when it is first turned on. You can initialize the plotter by turning it on and off, pressing **RESET** on the front panel, or executing the initialize instruction, IN.

Scaling — dividing the plotting area into units convenient for your application.

Scaling Points — points that are assigned the user-unit values specified by the parameters of the scale instruction, SC. These points are known as P1 and P2, and define diagonally opposite corners of a rectangular area. The locations of the points can be defined with the input P1 and P2 instruction, IP.

User Units — the units that you specify with the scale instruction, SC.

Plotter Units — the fixed units that the plotter understands. Each plotter unit is 0.025 mm; alternatively, 40 plotter units = 1 mm, or 1016 plotter units = 1 in.

NOTE: For more information on the basic concepts of scaling, refer to the Scaling, User Units, and Plotter Units discussions in Chapter 2. ■

HP-GL Syntax

Each HP-GL instruction begins with a two-letter mnemonic, which may be upper- or lowercase. If parameters are required following the mnemonic, they must be separated from each other by at least one comma or space.

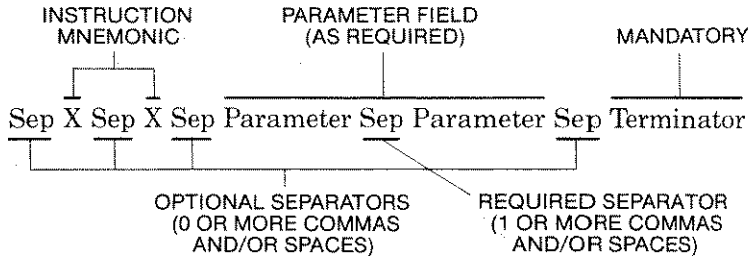
In both the HP-IB and RS-232-C interface configurations, HP-GL instructions are terminated by a semicolon or the first letter of the next mnemonic. In the HP-IB configuration, HP-GL instructions are also terminated by a line feed.

In either interface configuration, carriage-return characters are ignored, except as label characters or as an output response terminator in an RS-232-C configuration. The label instruction, LB, buffer label instruction, BL, and write to display instruction, WD, are special cases: each

ManualsPlus.com

must be terminated with the label terminator character. This character defaults to the ASCII end-of-text character, **ETX** (decimal code 3), but may be changed from its default value using the define terminator instruction, DT.

Only the terminator and the separators between parameters are required. Other separators may be inserted, as shown in this diagram.



To help you learn to recognize the mnemonics and their separators, the syntax examples in this manual do not add separators between the letters of the mnemonic. Also, parameters are separated by commas (but you can use spaces instead of commas).

Omitting Optional Parameters

Some instructions have optional parameters which, when omitted, assume a default value. In order to omit a parameter, **all subsequent parameters in the same instruction must be omitted**. (The only exceptions are the pen control parameters in the user-defined character instruction, UC, and the define downloadable character instruction, DL.)

For example, the fill type instruction, FT, has three optional parameters. If all three are included, the usual way to send FT to the plotter might be:

```
FT 3,100,45;
```

(In this example, commas are used as parameter separators. Spaces may be used instead.) If you were to omit the second parameter, you must also omit the third parameter. Thus, you would send:

```
FT3;  (not FT 3,,45; )
```

ManualsPlus.com

Parameter Formats

The parameter fields must be specified in the format defined by each respective HP-GL instruction (as shown in the syntax table accompanying each instruction's description throughout this manual). The format can be of three types:

1. Integer: The parameter must be an integer between -2^{23} and $2^{23}-1$ after rounding ($2^{23} = 8\,388\,608$). If the parameter must be an integer and you specify a decimal number, rounding is automatically performed. If the first digit after the decimal point is ≥ 5, positive parameters are incremented by 1 and negative parameters are decremented by 1. If no sign is specified, the parameter is assumed to be positive; e.g., 1008.5 is rounded to 1009 while −1008.6 is rounded to −1009.

2. Decimal: A number where the integer portion is between -2^{23} and $2^{23}-1$, and the optional decimal fraction has a maximum of 8 significant digits. The decimal point may be omitted when no decimal fraction is specified. If no sign is specified, the parameter is assumed to be positive.

   NOTE: If a parameter can be in decimal format and scaling is **on** (i.e., you are using user units), the plotter uses the full decimal fraction (up to 8 digits) for moving the pen. However, if a parameter can be in decimal format and scaling is **off** (i.e., you are using plotter units), the plotter will only move the pen to the point indicated by the integer portion of the decimal parameter. This is because the smallest move the plotter can make is 1 plotter unit (not a fraction of a plotter unit). However, note that the plotter still monitors the pen position using the complete decimal number (including the fractional portion). This can affect future pen positions, particularly if you are specifying a series of relative pen moves. For example, if you tell the plotter to move to the point 0.6,0.6 relative to the current pen position, the pen will not move (the integer position is 0,0). However, if you next tell the plotter to make an additional move to the relative point 0.7,0.7, the pen will move. The monitored cumulative move is to the point 1.3,1.3 (0.6 + 0.7 = 1.3), but the pen will move only to the point indicated by the integer portion, i.e., 1,1. ■

3. Label: Any sequence of characters. Refer to The Label Instruction, LB, (Chapter 7) for a complete description.

ManualsPlus.com

Notations Used in This Manual for Expressing Syntax

The syntax shown under the description for each HP-GL instruction uses the following notations:

*MN*emonic	— For readability, the mnemonic is shown upper-case and separated from the parameters and/or terminator.
required parameter	— All typeset items are required parameters.
( )	— All items in parentheses are optional.
c...c	— Any number of labeling characters.
(,...)	— Any number of the specified parameter (there must be an even number of X,Y coordinates in order to have complete X,Y coordinate pairs).
term	— Required terminator. A semicolon or the next mnemonic are valid terminators. For HP-IB configurations, a line feed (**LF**) is also valid.
[TERM]	— The output terminator. For HP-IB configurations, all output responses include a carriage return and line feed [**CR LF**] as a terminator. For RS-232-C configurations, all output responses include a terminator as defined by the set output mode instruction, ESC . M (refer to Chapter 16). The default output terminator is a carriage return [**CR**].

NOTE: The output terminator, denoted by [TERM] is sent from the plotter to the computer at the end of a response to an output instruction. This differs from the HP-GL terminator, denoted by *term*, which indicates the end of an HP-GL instruction sent to the plotter from the computer. You will learn more about output instructions and output terminators in Chapter 13. ■

Compatibility with Syntax of Other HP-GL Plotters

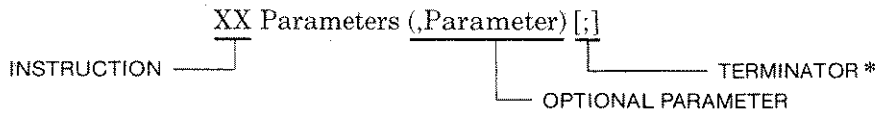
The syntax rules implemented on the HP 7550 are extremely flexible and backwards compatible with the HP 7470, 7475, 7580, 7585, and 7586 graphics plotters. However, if you are interested in writing software to drive all of these plotters, you should note the minor differences in syntax, as listed next.

- Separators — The HP 7550 and all of the above plotters allow a comma or space as a separator. However, the HP 7470 and 7475 both additionally allow a + or − sign as a separator.

ManualsPlus.com

- Terminators — The HP 7550 allows the same terminators as the above plotters.

Also note that the flexible syntax of all of the above plotters will not be able to drive earlier plotters such as the HP 9872. If you wish to write software that will drive earlier HP plotters, you should use the more rigorous syntax of the HP 9872, which is shown next.



The HP 9872 syntax totally ignores spaces. It does not allow commas between the characters of the mnemonic; only one comma must separate parameters, and only a semicolon or, in the HP-IB environment, a **LF** may be used as the terminator. In addition, parameters requiring integer format may not contain a decimal point or decimal fraction.

NOTE: The HP 7550 plotter implements an expanded HP-GL instruction set. Thus, some HP plotters may not respond to all of the instructions that the HP 7550 implements. In addition, some parameters and parameter formats may be interpreted slightly differently. ■

A Note about HP-GL Errors

If you execute an instruction that has errors in syntax or parameters, the plotter informs you by displaying an error message on the front panel. The error message remains on the front-panel display until you press a front-panel key or read the error number using the output error instruction, OE (presented in Chapter 13).

The following table provides a summary of the types of conditions that can cause an error. (This table also appears in Appendix B.) In addition, the descriptions of each HP-GL instruction throughout this manual provide tables that show the conditions that are most likely to occur with the particular instructions. Note that sending an instruction without parameters never causes an error condition. Instead, the instruction will either be ignored or it will be executed with default parameters, depending on the particular instruction.

*In an HP-IB environment, a LF may also be used for the terminator.

ManualsPlus.com

HP-GL Errors

Error	Displayed Message	Usual Plotter Reaction	Possible Cause
1	Command not recognized	Ignores the instruction	A mnemonic is incorrect or missing; an alphabetic character was specified in a parameter when a numeric character was expected.*
2	Wrong number of parameters	If too few parameters, ignores the instruction. If too many parameters, executes the instruction with the correct number of parameters and ignores the rest.	Too few or too many parameters; an incomplete X,Y coordinate pair.
3	Bad parameter	Ignores the instruction	A parameter is out-of-range.**
4	(unused)	(unused)	(unused)
5	Unknown character set	Ignores the instruction	A set other than −1, 0–9, 10–19, 30–39, or 40–49 has been designated or invoked.
6	Position overflow	Ignores the instruction	A single label is so long that it exceeds the plotter's numeric range.

*Check for typographical errors in the mnemonic and the parameters (a common mistake is to type O for 0, or vice versa). Also, check to be sure instructions are not being terminated prematurely. For example, if you are using an HP-IB configuration, check to be sure the computer is not inserting line feeds (LF) in the middle of a string of instructions. If an instruction is terminated after the mnemonic and before the parameters, the subsequent receipt of the parameters without a mnemonic will cause error 1.

**Check each parameter range for the instructions that are suspected of causing the error. Parameter ranges are listed at the beginning of each instruction's discussion throughout this manual, as well as in Appendix C.

(Table continued)

ManualsPlus.com

HP-GL Errors (Continued)

Error	Displayed Message	Usual Plotter Reaction	Possible Cause
7	Buffer overflow	Excecutes the data that fits in the affected buffer and ignores the data that overflows	One of the graphics memory buffers does not have enough space allocated.***

***Check buffer allocations. Refer to The Allocate Configurable Memory Instruction, ESC . T, in Chapter 14 for information on the plotter's buffers.

The Default Instruction, DF

USES: The DF instruction sets certain plotter functions to predefined default conditions. Use this instruction to return certain HP-GL instructions to a known state while maintaining conditions that may have been set from the front panel. This assures that unwanted graphics parameters such as scaling or character size are not inherited from another program, but that operator-set conditions such as pen speed and force remain unchanged.

SYNTAX: *DF term*

EXPLANATION: No parameters are used. However, the terminator must be included to complete the instruction. If parameters are included, error 2 is generated and the DF instruction is executed anyway.

The plotter functions affected by the DF instruction are listed in the following table.

Function	Equivalent Instruction	Default Condition
Pen control	AP;	Automatic as follows: • Lift or store a motionless pen after 15 seconds for transparency fiber-tip pens or drafting pens, or after 65 seconds for paper fiber-tip pens and roller-ball pens • Select pen only when required to draw
Label buffer	BL ETX	Cleared
Alternate set	CA 0;	Character set 0

(Table continued)

ManualsPlus.com

Function	Equivalent Instruction	Default Condition
Character selection mode	CM;	HP 7-bit mode
Standard set	CS 0;	Character set 0
Chord tolerance	CT;	Set to angle mode for AA, AR, CI, and WG instructions
Character chord	CC;	Set variable-space font chord angle to 5 degrees
Digitize clear	DC;	Clear DP instruction and return to current display
Downloadable character buffer	DL;	Cleared
Relative direction	DR 1, 0;	Horizontal characters
Label terminator	DT;	**ETX** (decimal code 3)
Extra space	ES 0, 0;	No extra space between characters
Fill type, spacing, and angle	FT;	• Type 1, solid bidirectional fill • 1% of the diagonal distance between P1 and P2 • 0 degrees
Mask value	IM 223, 0, 0;	Recognizes all defined errors
Input window	IW;	Set to hard-clip limits
Label origin	LO 1;	Standard labeling starting at current position
Line type and pattern length	LT;	• Type 1, solid line • 4% of the diagonal distance between P1 and P2
Plotting mode	PA;	Absolute
Polygon mode	PM 0; PM 2;	Polygon buffer cleared
Pen thickness	PT;	0.3 mm
Scaling	SC;	User-unit scaling off
Character slant	SL 0;	0 degrees
Symbol mode	SM;	Off

(Table continued)

ManualsPlus.com

Function	Equivalent Instruction	Default Condition
Relative size	SR;	• Character width = 0.75% of $\lvert P2_X - P1_X \rvert$ • Character height = 1.5% of $\lvert P2_Y - P1_Y \rvert$
Select set	SS;	Select standard character set
Tick length	TL;	tp = tn = 0.5% of $\lvert P2_X - P1_X \rvert$ for Y-tick and 0.5% of $\lvert P2_Y - P1_Y \rvert$ for X-tick
User-defined fill type	UF;	Solid bidirectional fill

The following conditions are also established:

- The carriage-return point for labeling instructions is updated to the current pen position.
- PD and PU instructions with parameters are defaulted to be forms of the PA instruction.
- Although character size is defaulted as if "SI;" were executed, subsequent changes to the scaling points P1 and P2 will cause the character size to vary as if "SR;" were executed.

The following plotter functions are **not** affected by a DF instruction:

- Locations of P1 and P2
- Current pen and its position
- Pen speed, force, and acceleration
- 90-degree rotation or axis alignment
- Curved line generator (CV instruction)
- Setting of these front-panel conditions: **AUTO FEED** key, standard/enhanced, HP-IB address, eavesdrop/standalone, handshake, modem/direct, full/half duplex, parity, 7-bit/8-bit, and baud rate
- Function key definitions established by the KY and WD instructions

ManualsPlus.com

The Initialize Instruction, IN

USES: The IN instruction returns most plotter settings to their initial power-on conditions. Use this instruction to return the plotter to a known state and to cancel most conditions that may have been set from the front panel. In this manual, the program examples begin with IN to clear unwanted graphics parameters from the previous program. The IN instruction is equivalent to pressing **RESET** on the front panel. This instruction has no effect on handshake protocol in any RS-232-C environment, but does clear any existing I/O error condition. It also has no effect on buffer sizes changed by the ESC.T or GM instruction. Any program that changes buffer sizes should return them to their defaults at the end of the program.

SYNTAX: *IN term*
or
IN −1 term

EXPLANATION: If parameters other than −1 are included, error 2 is generated and the IN instruction is executed anyway.

The IN instruction sets the plotter to the same conditions as the default instruction, DF, and sets these additional conditions:

- Raises the pen graphically and physically.
- Cancels 90-degree rotation.
- Sets default pen speed, force, and acceleration values in accordance with the carousel installed in the plotter.
- Sets bit position 3 of the status word to 1 (to indicate the plotter has been initialized).
- Clears any HP-GL (graphics) error condition.
- Clears lost mode (refer to Relationship of Plotting Instructions and Graphics Limits in Chapter 4).
- Clears the display and removes any function key definitions established by the WD and KY instructions.
- Sets the group count to 0 (GC instruction).
- Turns off the curved line generator (CV instruction).
- Returns P1, P2, and the axis-align point to the X,Y coordinate values that were set when the paper limits were established. However, any existing axis alignment is maintained.

The IN −1 instruction resets all of the above conditions *except* the pen control (AP instruction) settings.

ManualsPlus.com

NOTE: If an axis alignment has been set, only P1 will return to its default physical position on the paper. The hard-clip limits are still compressed, and P2 and the axis-align point will be rotated from their default physical positions by an amount corresponding to the axis alignment. Axis alignment is described in the Operation and Interconnection Manual. ■

An IN instruction is equivalent to switching the plotter off and then on again, except that axis alignment is not changed and buffer sizes are not reset. Remember that the following front-panel functions are stored in the plotter's continuous memory; neither an IN instruction, nor switching power off and on, affects the current setting of these functions: **AUTO FEED** key, standard/enhanced, HP-IB address, eavesdrop/standalone, handshake, modem/direct, full/half duplex, parity, 7-bit/8-bit, and baud rate.

The Input P1 and P2 Instruction, IP

USES: The IP instruction allows you to establish new or default locations for the scaling points P1 and P2. P1 and P2 are used by the scale instruction, SC, to establish user-unit scaling. The IP instruction is often used to ensure that a plot is always the same size, regardless of how P1 and P2 might have been set from the front panel. This instruction can also be used in advanced techniques such as plotting mirror images, enlarging/reducing plots, and enlarging/reducing relative character size or direction (refer to Chapters 7 and 9).

SYNTAX: *IP* $P1_X$, $P1_Y$ (, $P2_X$, $P2_Y$) *term*
or
IP *term*

Parameter	Format	Range	Default
X- and Y-coordinates	integer	-2^{23} to $2^{23} - 1$ plotter units	depends on paper size

EXPLANATION: Issuing IP without parameters (IP ;) sets P1 and P2 to the default positions established by the current paper size, adjusted by any current axis alignment or coordinate-system rotation. The default coordinates of P1 and P2 when no coordinate-system rotation has occurred are listed in the following table. (If axis alignment has occurred, P1 and P2 might be adjusted by a few plotter units. Axis alignment can only be accomplished with the **ALIGN** function key; refer to the Operation and Interconnection Manual.)

ManualsPlus.com

Paper Size	Default (in Plotter Units)	
	$P1_X,P1_Y$	$P2_X,P2_Y$
A3	380,430	15580,10430
A4	430,200	10430,7400
A	80,320	10080,7520
B	620,80	15820,10080

NOTE: If an IP instruction is executed without parameters after the coordinate system has been rotated 90 degrees by the front-panel **ROTATE-90** function key or the RO90; instruction, the default locations of P1 and P2 are changed to reflect the rotation. The X- and Y-coordinates for each scaling point are reversed. For example, IP; after a 90-degree rotation on A-size paper sets P1 to be 320,80 and P2 to be 7520,10080. ■

Specifying the P2 X,Y coordinates is optional. However, if they are not specified, then P2 tracks P1 and its coordinates change so that the X- and Y-distances between P1 and P2 do not change. P2 could become located outside the hard-clip limits when this happens. (The axis-align point always tracks P1, and its coordinates change when P1 moves.) This tracking phenomenon is useful for preparing more than one equal-sized plot on one page. For an example, refer to Techniques for Changing the Plotting Area in Chapter 9.

If you specify both P1 and P2 coordinates, and either coordinate of P2 is set equal to the corresponding coordinate of P1, then the plotter increments that coordinate of P2 by one plotter unit. For example, suppose you specify IP 0,1000,500,1000;. The plotter increments the Y-coordinate of P2 as if you had specified IP 0,1000,500,1001;.

The usual orientation of P1 and P2 is for P1 to define the lower-left corner of a rectangular area ("lower-left" is with relation to the origin 0,0, which is in the lower-left corner of the page), and for P2 to define the upper-right corner. It is possible to specify coordinates such that P1 and P2 define different diagonal corners (e.g., P1 is lower-right and P2 is upper-left). If so, and scaling is on, the plot will be reversed and/or turned upside down (this is called a mirror image). These possibilities are discussed in Chapter 9. It is also possible to set P1 and P2 to be outside the hard-clip limits.

The IP instruction remains in effect until another IP instruction is executed, P1 and P2 are changed from the front panel, or the plotter is initialized.

ManualsPlus.com

The Scale Instruction, SC (presented next in this chapter), shows examples of changing the locations of P1 and P2. Many other examples throughout this manual also use IP to change P1 and P2.

The following table summarizes the possible error conditions or unexpected results that you might observe with the IP instruction.

Condition	Error	Plotter Response
no parameters	none	establishes default P1 and P2 for paper size, adjusted by any axis alignment or rotation
$P1_X = P2_X$ or $P1_Y = P2_Y$	none	increments $P2_X$ or $P2_Y$ by 1
1 or 3 parameters	2	ignores instruction
more than 4 parameters	2	executes first 4 parameters
parameter out-of-range	3	ignores instruction

The Scale Instruction, SC†

USES: The SC instruction establishes a user-unit coordinate system by mapping values onto the scaling points P1 and P2. Thus, you can plot in user units convenient to your application. In addition, you can use this instruction to establish automatic isotropic scaling or to relocate the origin and set a specific ratio of plotter units to user units. For a discussion of the concept of scaling, refer to Scaling in Chapter 2.

SYNTAX: *SC* X_{min}, X_{max}, Y_{min}, Y_{max} (, type (, left , bottom)) *term*

or

SC X_{min}, X_{factor}, Y_{min}, Y_{factor}, 2 *term*

Parameter	Format	Range	Default
X_{min}	decimal	-2^{23} to $2^{23}-1$	none
X_{max}	decimal	-2^{23} to $2^{23}-1$	none
Y_{min}	decimal	-2^{23} to $2^{23}-1$	none
Y_{max}	decimal	-2^{23} to $2^{23}-1$	none

(Table continued)

†This instruction applies to plotters whose serial number prefix is 2629A or higher. If the serial number prefix is 2520A or lower, refer to Appendix A for the SC instruction for your plotter.

ManualsPlus.com

Parameter	Format	Range	Default
type	integer	0 and 1	0
left	decimal	0 to 100%	50%
bottom	decimal	0 to 100%	50%
X_{factor}	decimal	0.1 to $2^{23}-1$	none
Y_{factor}	decimal	0.1 to $2^{23}-1$	none

EXPLANATION: The Scale instruction comes in two forms. The X_{min}, X_{max}, Y_{min}, Y_{max} parameters represent the X- and Y-axis ranges, respectively. As a result, the first and third parameters (X_{min} and Y_{min}) in these ranges are the coordinate pair that is mapped onto P1; the second and fourth parameters (X_{max} and Y_{max}) are the coordinate pair that is mapped onto P2. (This is different from the IP instruction, where the parameters are expressed as X,Y coordinate pairs rather than as ranges.)

As their names suggest, you will normally want to specify X_{min} smaller than X_{max}, and Y_{min} smaller than Y_{max}. However, it is permissible to specify X_{min} larger than X_{max} and Y_{min} larger than Y_{max}. (If you do this, your plot could be drawn as a mirror image — reversed and/or upside down — depending on the relative positions of P1 and P2.) X_{min} cannot be set equal to X_{max}, and Y_{min} cannot be set equal to Y_{max}.

The parameters of the SC instruction are always mapped onto the current P1 and P2 locations. P1 and P2 retain these new values until scaling is turned off or another SC instruction redefines the user-unit values. Thus, the size of a user unit could change if any change is made in the relative position and distance between P1 and P2 *after* an SC instruction is executed.

The optional type, left, and bottom parameters allow you to specify anisotropic or automatic isotropic scaling. Anisotropic scaling (type parameter = 0) is the default setting and allows a user unit to be a different size along the X-axis and the Y-axis. If your plot includes letters or shapes such as circles, they will be distorted if your scaling is anisotropic. For example, a circle will turn out as an ellipse (egg-shaped) rather than a circle.

Isotropic scaling means the user unit is the same size on the X- and the Y-axis. You can set isotropic scaling manually by setting P1 and P2 so that they define a square area (refer to Appendix A for more information on manual methods of establishing isotropic scaling). The easiest way, however, is to set it automatically using the type parameter. When the type parameter equals 1 and the plotter encounters a CI circle, PM

ManualsPlus.com

polygon, SR character size, DR label direction, LT pattern length, or TL tick length instruction — it will automatically create the largest square (isotropic) plotting area that will fit in the current P1,P2 plotting area and use the isotropic area for plotting.

Normally, the isotropic area is centered with the blank space equally divided between left and right or top and bottom; e.g., the defaults of the left and bottom parameters of 50%. You can however, move the isotropic area around in the P1,P2 plotting area using the left and bottom parameters. The left parameter indicates the percentage of the blank space on the left of the isotropic area; the bottom parameter indicates the blank space below the isotropic area. For example, SC 0,500,0,1000,1,0,0 would establish P1 as 0,0, P2 as 500,1000, isotropic scaling, and locate the plotting area in the bottom left-hand corner of the P1,P2 plotting area.

The second form of the scaling instruction allows you to shift the plotter-unit origin (0,0) relative to P1 while maintaining a specific ratio of plotter units to user units. The position of P2 is irrelevant with this form of scaling. X_{min} and Y_{min} are the user-unit coordinates of P1. X_{factor} represents the number of plotter units per user unit on the X-axis; Y_{factor} represents the number of plotter units per user unit on the Y-axis. For example, SC 0,40,0,40,2 allows metric scaling since 1 millimetre = 40 plotter units. Each user unit will be 1 millimetre. SC 0,1.016,0,1.016,2 allows scaling in thousandths of an inch since 1 inch = 1016 plotter units. Each user unit will be 0.001 of an inch. SC 600,1,600,1,2 shifts the plotter-unit origin to the lower-left corner while maintaining user units the same size as plotter units. This allows you to plot to the lower-left corner without using negative coordinates.

While scaling is on (i.e., after an SC instruction has been executed), only those plotting instructions that can be issued in "current units" are interpreted as user units; all instructions that can only be issued in plotter units are still interpreted as plotter units. (The table in the SYNTAX section of each instruction tells you what kind of units each parameter requires.)

An SC instruction without parameters (SC;) turns off user-unit scaling; all subsequent plotting instructions will then be interpreted in plotter units.

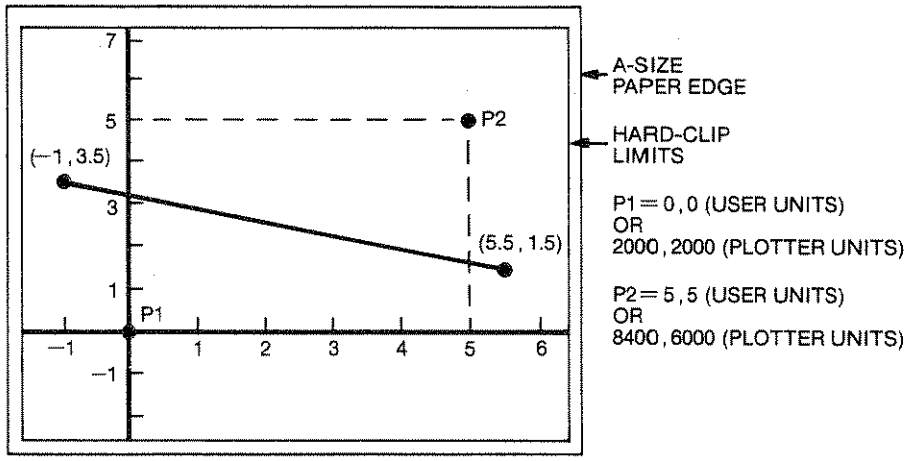
The table on the next page summarizes the possible error conditions or unexpected results that you might observe with the SC instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	turns scaling off
more than 7 parameters	2	executes first 4 parameters
less than 4 parameters	2	ignores instruction
$X_{min} = X_{max}$ or $Y_{min} = Y_{max}$ type other than 0 or 1 or parameter out-of-range	3	ignores instruction

Remember that the SC parameters are mapped onto the current locations of P1 and P2. P1 and P2 do not represent a graphic limit; therefore, the new user-unit coordinate system extends across the entire range of the plotter-unit coordinate system. Thus, you can plot to a point beyond P1 or P2, as long as you are within the hard-clip limits. For example, you can plot from the point −1 , 3.5 to the point 5.5 , 1.5 as illustrated below.

```
"IP 2000,2000,8400,6000;"
"SC 0,5,0,5;"
```



ManualsPlus.com

The Graphics Memory Instruction, GM

USES: The GM instruction allocates memory to four of the five separate buffers in the configurable graphics memory. Use this instruction to change buffer sizes when your program includes any of the following HP-GL instructions: polygon mode instruction, PM; downloadable character instruction, DL; replot instructions, BF and RP; curved line generator instruction, CV.

SYNTAX: *GM* (polygon buffer) (, downloadable character buffer) (, replot buffer) (, vector buffer) *term*

Parameter	Format	Range*	Default
polygon buffer	integer	4–12 752 bytes	1778
downloadable character buffer	integer	0–12 754 bytes	0
replot buffer	integer	0–12 750 bytes	9954
vector buffer	integer	44–12 794 bytes	44

*Range values assume the default I/O buffer size of 1024 bytes.

EXPLANATION: The configurable graphics memory is an area in the plotter reserved for temporary data storage. The memory can store up to 12 800 bytes of information. Storage is divided between five buffers: physical I/O buffer, polygon buffer, downloadable character buffer, replot buffer, and vector buffer.

You can change buffer sizes with either of two instructions: the graphic memory instruction, GM, or the allocate configurable memory instruction, ESC . T. The ESC . T instruction is a device-control instruction that allocates memory to all five buffers. Each of the buffers is described in detail under The Allocate Configurable Memory Instruction, ESC . T, in Chapter 14. Refer to these descriptions for information about GM parameters.

In comparing the GM and ESC . T instructions, note that GM first clears all the buffers, then allocates memory to *four* of the five buffers; unlike ESC . T, the GM instruction does not change the size of the physical I/O buffer. The number of bytes available to the remaining four buffers depends on the current size of the physical I/O buffer. (Unless you or someone else changes memory allocations, the physical I/O buffer will be set at its default value, 1024 bytes.)

The combined default values of all five buffers equals the total storage capacity of the configurable memory, 12 800 bytes. So, when you increase one buffer above its default value, you must "borrow" bytes from another or you will exceed the memory limits. When this happens, the plotter uses an algorithm to reduce the largest parameters. (The algorithm is described at the end of the ESC . T instruction in Chapter 14.)

ManualsPlus.com

Each parameter in the GM instruction reflects the number of bytes assigned to a specific buffer. A buffer must remain within the size range listed in the previous table. If you specify fewer bytes than the minimum, the plotter allocates the minimum. If you specify more bytes than the maximum, the plotter will perform the algorithm and reduce the buffer's size accordingly. When a GM parameter is omitted, the plotter assigns the default value to the corresponding buffer. A GM instruction without parameters sets the four buffers to their default values.

The memory allocations you assign remain in effect until one of the following conditions occurs:

- The plotter is turned off. When the plotter is turned on again, the buffers are automatically cleared and set to default values.
- The plotter receives the reset instruction, ESC.R. This instruction clears all buffers and defaults buffer allocations.
- The plotter receives a GM or ESC.T instruction with different memory allocations. Each new memory allocation instruction clears the buffers before allocating memory.

The following table summarizes the possible error conditions or unexpected results that you might observe with the GM instruction.

Condition	Error	Plotter Response
too many parameters	2	ignores extra parameters
parameters' sum exceeds 12800 bytes	3	performs algorithm and reduces parameters accordingly
buffer overflow	7	executes data that fits in affected buffer and ignores the data that overflowed

Hints for Using the GM Instruction

Send GM at the beginning of a program, before you send the HP-GL instructions that draw your plot. If the buffers receive plot information before memory has been allocated, all data stored in them will be lost when they are cleared by GM.

Allocate memory with GM when your computer cannot read an output statement; otherwise, allocate memory with ESC.T. The ESC.T instruction allows you to control the size of all five buffers. However, to avoid losing data, when you send ESC.T you must also precede and follow it with ESC.L. The ESC.L instruction generates an output statement that must be read by your computer.

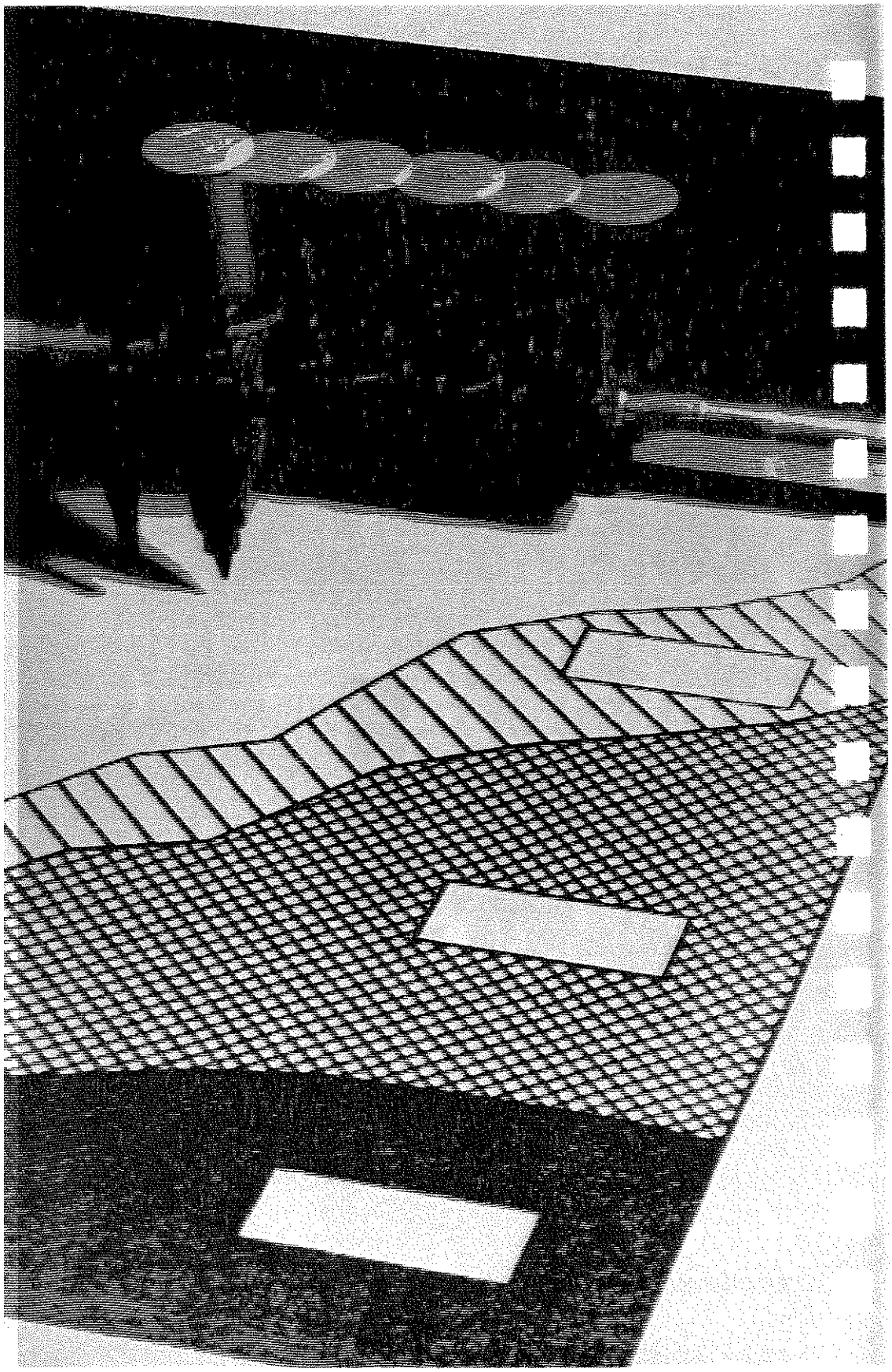
ManualsPlus.com

To check memory allocations, send the output configurable memory size instruction, ESC.S, described in Chapter 14. The ESC.S instruction outputs the total number of bytes allocated in the configurable graphics memory, or the number of bytes in any of its buffers.

ManualsPlus.com

Notes

ManualsPlus.com



ManualsPlus.com

Chapter 4
Pen Control and Plotting

What You'll Learn in This Chapter

Now that you understand the unit systems in which data can be represented, you are ready to create plots. In this chapter, you will learn how to select or change pens, how to raise and lower the pen, and how to plot lines. You will also learn what happens when you try to plot beyond the graphics limits, and how to send variables as part of a plotting instruction.

HP-GL Instructions Covered

SP The Select Pen Instruction
PU The Pen Up Instruction
PD The Pen Down Instruction
PA The Plot Absolute Instruction
PR The Plot Relative Instruction

Terms You Should Understand

Point — an X,Y coordinate pair.

Absolute Plotting — plotting to a point whose location is specified with respect to the fixed origin (0,0). This concept is described in detail under Absolute and Relative Movement in Chapter 2.

Relative Plotting — plotting to a point whose location is specified relative to the current pen position. This concept is described in detail under Absolute and Relative Movement in Chapter 2.

Current Units — plotter units (if scaling is off) or user units (if scaling is on). Scaling is on when an SC instruction with parameters has been executed; otherwise, scaling is off. For discussions of plotter units, user units, and scaling, refer to Chapters 2 and 3. The parameters of some plotting instructions can be interpreted as plotter units or user units (depending on whether scaling is off or on). If this is the case, the descriptions of parameter ranges in this manual indicate "current units."

ManualsPlus.com

Constant — a number that has a fixed value. The coordinates of the point 10 , 20 are constants.

Variable — a symbol that represents a value and can be changed. The coordinates of the point X,Y are variables. To define the values for the variables, you would need expressions such as X = 25 and Y = 35.

The Select Pen Instruction, SP

USES: The SP instruction moves pens between the carousel and the pen holder. Use this instruction to load a pen into the pen holder so that drawing will occur. You can also use it to select pens of different colors or widths during a plotting program. Then, use SP at the end of a program to return the pen to the carousel.

SYNTAX: *SP* pen number *term*
or
SP *term*

Parameter	Format	Range	Default
pen number	integer	0-8	0

EXPLANATION: The pen number parameter corresponds to the pen numbers marked on the carousels. The pen number is interpreted as follows:

1. Pen Number = 1-8. Pen numbers 1 through 8 cause the appropriate pen to be placed in the pen holder. If there is currently a pen in the holder, this pen is stored before the new pen is selected. When selecting a pen, the SP instruction is ignored in these cases: there is no pen in the selected stall, a pen number larger than 8 is selected, or there is no carousel in the plotter. After selecting a new pen, the pen holder returns to the current graphics position.

2. Pen Number = 0 or No Parameter. Pen number 0 (or no parameter) returns the pen currently in the pen holder to the stall from which it came, if possible. If the stall is occupied, the pen is placed in the vacant stall with the lowest number. The SP ; instruction is ignored if there are no vacant stalls, or if there is no carousel in the plotter. After storing a pen, the pen holder remains by the carousel, but the current graphics position does not change.

An SP instruction remains in effect until the plotter is initialized, a new pen is selected, or the current pen is returned to a stall in the carousel. Pens can be selected and returned by using the front-panel controls or by executing valid SP instructions.

ManualsPlus.com

NOTE: When bit 2 of the pen control word is set (refer to The Automatic Pen Operations Instruction, AP, in Chapter 10), the pen is not selected in response to an SP instruction until it is needed to draw. This is the case when the plotter is first turned on or set to default conditions with the DF or IN instructions. ■

The following table summarizes the possible error conditions or unexpected results that you might observe with the SP instruction.

Condition	Error	Plotter Response
no parameter	none	returns pen to stall, if possible
pen number less than 0	3	ignores instruction
pen number greater than 8	none	ignores instruction

The Pen Instructions, PU and PD

USES: The pen up instruction, PU, raises the pen; the pen down instruction, PD, lowers the pen. Use PU or PD with parameters to move or draw to the points specified by the parameters.

SYNTAX: *PU* X-coordinate, Y-coordinate (,...) *term*
or
PU *term*

and

PD X-coordinate, Y-coordinate (,...) *term*
or
PD *term*

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none

EXPLANATION: The PU and PD instructions are interpreted as follows:

1. X- and Y-coordinates. When parameters are included, PU and PD are interpreted as forms of one of the plot instructions, PA and PR, depending on which (PA or PR) was the most recent plot instruction. (If no plot instruction has been previously executed, PA is assumed.)

   For example, if you execute PD X, Y; and absolute plotting is in effect, the X,Y coordinates are interpreted as absolute coordinates.

ManualsPlus.com

Similarly, if relative plotting is in effect, PD X , Y ; causes the plotter to draw to the relative point X,Y. For complete information on how the parameters are interpreted, refer to The Plot Absolute Instruction, PA, and The Plot Relative Instruction, PR, located next in this chapter.

2. No Parameters. The PU instruction without parameters raises the pen without moving the pen to a new location. Similarly, the PD instruction without parameters lowers the pen without moving the pen to a new location. However, the pen is physically lowered only if it is within the current graphics limits (window) and is not in the "up" portion of a dashed line pattern. Executing a PU or PD instruction without parameters is the same as pressing the **PEN-UP** or **PEN-DN** function keys on the front-panel display.

The Plot Absolute Instruction, PA

USES: The PA instruction establishes absolute plotting mode and moves the pen to the specified point(s). You can use this instruction with PU to move to a point with the pen up, or with PD to plot to a point with the pen down.

SYNTAX: *PA* X-coordinate, Y-coordinate (,...) *term*
or
PA *term*

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none

EXPLANATION: The PA instruction is interpreted as follows:

1. X- and Y-coordinates. The X- and Y-coordinates specify the absolute position to which the pen will move. Both coordinates of each pair must be specified, and you may specify as many coordinate pairs as you wish. (This is limited only by the ability of your controller to output long strings.)

   When you include more than one coordinate pair, the pen moves to each point in the order given, using the current up/down status. The up/down status is based on the most recent PU or PD instruction. This concept is discussed in the next section, titled Plotting with PA, PU, and PD.

   The coordinates are interpreted as user units if scaling is on, or as plotter units if scaling is off.

ManualsPlus.com

2. No Parameters. The PA instruction without parameters establishes absolute plotting mode for subsequent PD or PU forms of the PA instruction.

The following table summarizes the possible error conditions or unexpected results that you might observe with the PA instruction.

Condition	Error	Plotter Response
no parameters	none	establishes plot absolute mode
odd number of coordinates	2	executes each coordinate pair; ignores extra coordinate
parameter out-of-range	3	ignores wrong coordinate and the rest of the instruction
lost mode	none	conforms to the conditions described under Relationship of Plotting Instructions and Graphics Limits later in this chapter

Plotting with PA, PU, and PD

As mentioned previously, the PA instruction moves the pen to the specified point(s) using the current pen up/down status. At power-on, after a front-panel reset, and after an IN instruction, the pen is up. Therefore, the following string of instructions will cause the pen to move to the point 2000,2000 without drawing a line.

```
"IN;SP1;PA2000,2000;"
```

Now, in order to draw from that point to another point, you must put the pen down. The following instructions draw a line from 2000,2000 to 5000,5000.

```
"IN;SP1;PA2000,2000;PD5000,5000;"
```

Note that PD was used to do absolute plotting. This is equivalent to issuing these separate instructions:

```
"IN;SP1;PA2000,2000;PD;PA5000,5000;"
```

You can also insert PD or PU within a list of X,Y coordinate pairs. When you do this, commas or spaces are required between numeric parameters, but are optional before and after the two-letter mnemonics. You must include the required terminator after the last entry. In the following example, the comma is used to indicate the optional separators and the semicolon is used to indicate the required terminator.

ManualsPlus.com

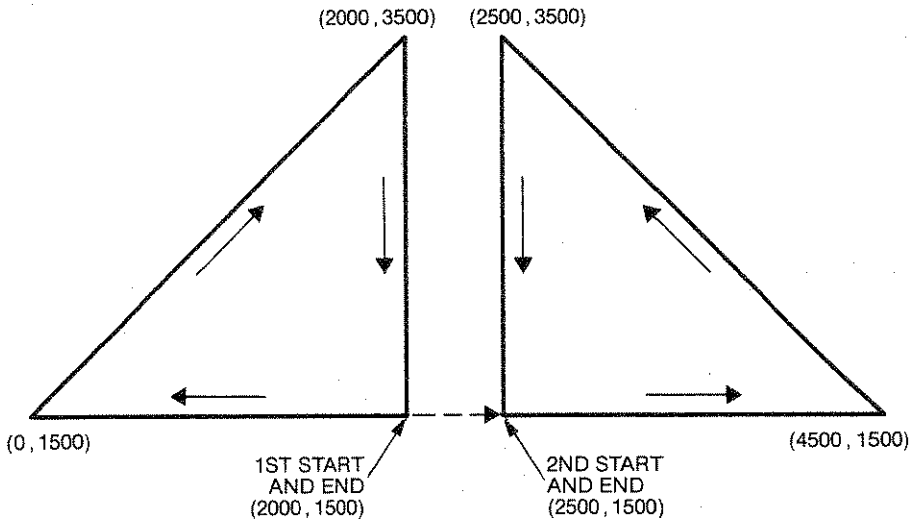
PA , X_1 , Y_1 , PD , X_2 , Y_2 , X_3 , Y_3 , PU , X_4 , Y_4 ;

OPTIONAL

The following BASIC program illustrates the concepts of using PA, PD, and PU to draw two triangles. The resulting plot is also shown, along with coordinates and arrows showing pen direction.

Line 10 of the program identifies the plotter as the system printer; this is because the PRINT statement is used to send the strings of HP-GL instructions to the plotter. Change the first line and the PRINT statements as necessary for your computer. Before running the program, be sure paper is loaded in the plotter.

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA2000,1500;"
30  PRINT #1, "PD0,1500,2000,3500,2000,1500;"
40  PRINT #1, "PU2500,1500;"
50  PRINT #1, "PD4500,1500,2500,3500,2500,1500;"
60  PRINT #1, "SP0;"
70  END
```



The Plot Relative Instruction, PR

USES: The PR instruction establishes relative plotting mode and moves the pen to the specified point(s), relative to the current pen position. You can use this instruction with PU to move to a point with the pen up, or with PD to plot to a point with the pen down.

ManualsPlus.com

SYNTAX: *PR* X-increment, Y-increment (,...) *term*
or
PR *term*

Parameter	Format	Range	Default
X- and Y-increments	decimal	-2^{23} to $2^{23}-1$ current units	none

EXPLANATION: The PR instruction is interpreted as follows:

1. X- and Y-increments. The X- and Y-increments specify the position to which the pen will move, relative to the current pen position. The X-increment specifies the number of units for the pen to move in the direction of the X-axis; the Y-increment specifies the number of units for the pen to move in the direction of the Y-axis. The signs of the increments determine the relative direction in which the pen moves; a positive value moves the pen to the right for X-increments and up for Y-increments, and a negative value moves the pen to the left for X-increments and down for Y-increments. Refer to Absolute and Relative Movement in Chapter 2 for details on relative plotting using increments.

   Both increments of each pair must be specified, and you may specify as many increment pairs as you wish. (This is limited only by the ability of your controller to output long strings.)

   When you include more than one increment pair, the pen moves to the first point using the current up/down status. This point then becomes the current pen position for the next increment pair. In this way, the pen moves to each point in the order given, still using the current up/down status. The up/down status is based on the most recent PU or PD instruction. This concept is discussed in the next section, titled Plotting with PR, PU, and PD.

   The increment parameters are interpreted as user units if scaling is on, or as plotter units if scaling is off.

2. No Parameters. The PR instruction without parameters establishes relative plotting mode for subsequent PD or PU forms of the PR instruction.

The following table summarizes the possible error conditions or unexpected results that you might observe with the PR instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	establishes plot relative mode
odd number of increments	2	executes each increment pair; ignores extra increment
parameter out-of-range	3	ignores wrong coordinate and the rest of the instruction
lost mode	none	conforms to the conditions described under Relationship of Plotting Instructions and Graphics Limits later in this chapter

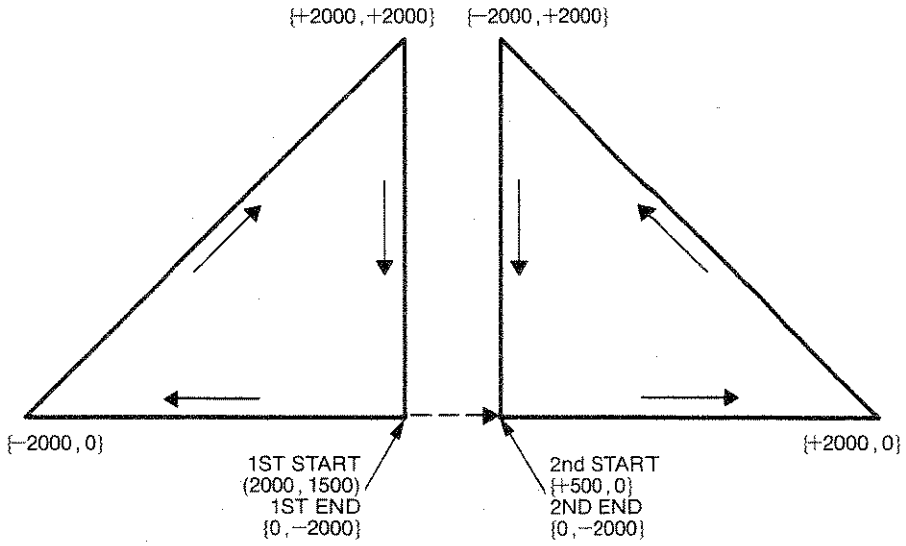
Plotting with PR, PU, and PD

The PR instruction operates with PU and PD in the same manner as described in this section under The Plot Absolute Instruction, PA. The only difference is that PR moves are relative to the current pen position, whereas PA moves are absolute with respect to the origin.

The following BASIC program illustrates this difference by using PR to draw the same triangles previously drawn with PA. Again, change line 10 and the PRINT statements as necessary for your computer. Also, be sure paper is loaded in the plotter before running this program.

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA2000,1500;"
30  PRINT #1, "PR;PD-2000,0,2000,2000,0,-2000;"
40  PRINT #1, "PU500,0;"
50  PRINT #1, "PD2000,0,-2000,2000,0,-2000;"
60  PRINT #1, "SP0;"
70  END
```

ManualsPlus.com



NOTE: Absolute coordinates are shown in parentheses, as in (2000,1500). Relative increments are shown in braces, as in {0,−2000}.

Relationship of Plotting Instructions and Graphics Limits

For most plotting situations, the pen will probably draw as you expect it to. However, depending on what your graphics limits (windows) are and the X,Y coordinates you specify, the pen might not always move and plot. This section describes the situations that affect the movement of the pen. It applies directly to these plotting instructions: PA, PR, RA, RR, EA, ER, CI, AA, AR, WG, EW, EP, FP, CP, LB, PB, UC, XT, and YT.

Types of X,Y Coordinates

The pen motion caused by plotting instructions depends on the type of X,Y coordinates specified. There are three types of X,Y coordinates:

1. Inside window area
2. Outside window area and in-range
3. Out-of-range

Out-of-range coordinates are defined as either X or Y, or both, being less than -2^{23} or greater than $2^{23}-1$. Specifying out-of-range coordinates sets error 3 (bad parameter) and results in the plotting instruction being ignored. (In a PA, PR, PU, or PD instruction, where you can

ManualsPlus.com

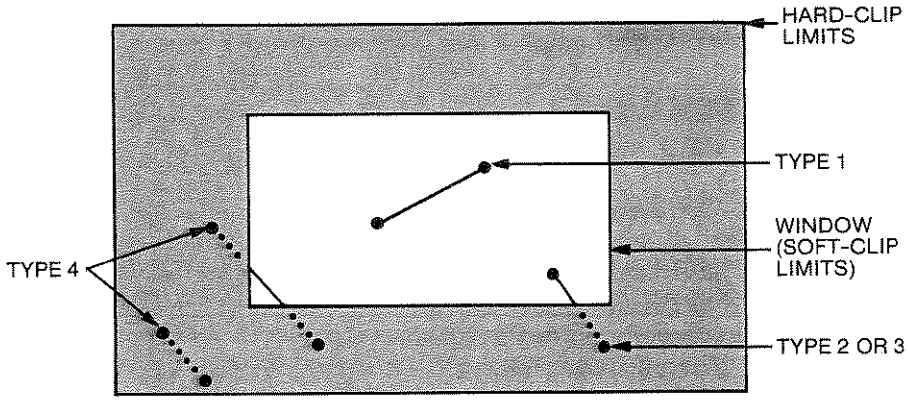
specify multiple coordinate pairs, only the pair with the out-of-range coordinate and all subsequent pairs are ignored; the previous pairs are still executed.)

The other two types of coordinates (inside window and outside window) are discussed below. Remember that plotting occurs only within the currently defined window. At power-on, the currently defined window is the hard-clip limits. However, you can define smaller windows (soft-clip limits) using the input window instruction, IW (refer to Chapter 9).

Plotting Inside and Outside of Windows

There are, in general, four types of line segments that can be specified from a given point to some new point, as follows:

	Last Point		New Point
1.	Inside window area	to	inside window area
2.	Inside window area	to	outside window area
3.	Outside window area	to	inside window area
4.	Outside window area	to	outside window area



In type 1, the pen moves from the last point to the new point with the pen up or down as programmed.

In type 2, the pen moves from the last point toward the new point and stops where the line segment between these points intersects the window limit. The pen up/down state is as programmed until the intersection point is reached. Then, the pen lifts and remains in this position until a plotting instruction is received that brings the pen back inside the window (refer to type 3).

In type 3, the pen movement depends on whether the pen is programmed to be up or down. If the pen is programmed to be up, it moves directly to the new point. If the pen is programmed to be down, it moves, up, to the point where the line segment between the last point and the new point intersects the window limit. When the pen reaches

ManualsPlus.com

this point, the pen assumes its programmed down position. The pen then continues plotting to the new point.

In type 4, no pen movement occurs unless some part of the line segment between the last and new points intersects the window area. If part of the line is in the window area, the pen moves, up, to the first intersection point with the window. The pen then moves as programmed (up or down) to the second intersection point, where it lifts and stops moving. Thus, this situation is a combination of type 3, followed by type 1, followed by type 2. If none of the line segments lies within the window area, the X- and Y-coordinates of the current pen position are updated even though the pen does not move.

Lost Mode

You should be aware of a condition known as lost mode, although you will probably never encounter it under normal plotting situations. Lost mode is established when the monitored graphics position exceeds the plotter's numeric range of -2^{23} to $2^{23}-1$. When scaling is on, this can occur when the plotter-unit equivalents of the specified user-unit coordinates exceed the plotter's numeric range. When scaling is off, lost mode is not likely to occur because error 3 (bad parameter) is generated when plotter-unit coordinates exceed this range, and the plotting instruction is ignored. However, regardless of whether scaling is on or off, relative moves (as with the PR, AR, RR, ER, and PM instructions) can cause the plotter to enter lost mode. Remember that relative moves are added to the current pen position. Thus, when a relative move causes the pen position to exceed the plotter's range, lost mode is entered.

While the plotter is in the lost mode, the AA, AR, CI, CP, LB, PB, PR, UC, XT, YT, RA, RR, EA, ER, WG, EW, FP, and EP are all ignored until lost mode is cleared. If a PD or PU instruction is received while in lost mode, the current pen status is updated, but the pen remains physically up until lost mode is cleared. However, pressing the **PEN-DN** function key on the front panel both lowers the pen and updates the pen status.

Any of the following will clear lost mode:

- executing a PA instruction with in-range parameters (if PA coordinates are in-range but outside of the window, lost mode is cleared, but the pen does not move)
- executing an IN or RO instruction
- pressing any of the following function keys on the front panel: **RESET**, **P1**, **P2**, **ENTER P1**, **ENTER P2**, any cursor key, or **ALIGN**
- transition to not-ready state (with the NR instruction or the front-panel **VIEW** function key)

ManualsPlus.com

Plotting with Variables

In many plotting applications, you might wish to use variables for the parameters of an HP-GL instruction. (For example, you might want to set up a program for drawing a line chart, but be able to change the data points each time you plot the chart.) This is simple to do. Just remember these principles:

- The values of all parameters have the same restrictions (integer or decimal in a valid range) when sent as variables as they do when sent as constants.
- HP-GL mnemonics, their separators, and their terminators all must be sent to the plotter along with the variable parameters.

NOTE: There are several ways that you can send HP-GL instructions to the plotter. Also, specific methods for defining output format vary from computer to computer. However, the principles are applicable to any computer. ■

Methods for Sending Variable Parameters

The usual way to send an HP-GL instruction with constant parameters is to send the mnemonic, parameters, separators, and terminator as a literal string. (On many computers, literal strings are delimited by quotation marks.) As shown below, this results in a compact field of characters being sent to the plotter, which reduces activity on the interface and saves time.

```
Instructions sent:     PRINT #1, "PA10,20,80,90;"
Characters received: PA10,20,80,90;
```

If you were to send variable parameters in the same manner, as shown next, you would generate an error because the plotter would not recognize the variables. The plotter looks at the characters X and Y and tries to interpret them as an HP-GL mnemonic, but does not recognize them. (Assume that the variables have been defined as X = 80 and Y = 90.)

```
Instructions sent:     PRINT #1, "PA10,20,X,Y;"
Characters received: PA10,20,X,Y;  and error 1
                                   (instruction not recognized)
```

Thus, you cannot send variables to the plotter as part of a literal string. Following are two acceptable methods of sending variables to the plotter, along with the characters that the plotter receives. Of course, there are many acceptable methods, which depend on your computer's format and output statements. These are formatted for Microsoft® BASIC. (Again, assume in each case that the variables have been

ManualsPlus.com

defined as X = 80 and Y = 90.) Note that the mnemonic and the terminator are always sent as literal strings.

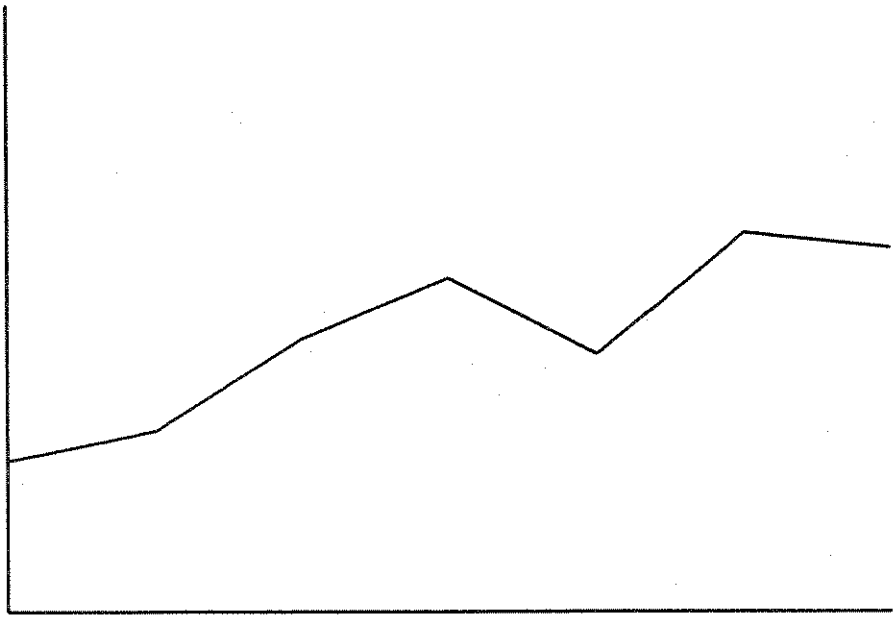
(1) Instructions sent: `PRINT #1, "PA10,20,",X,",",Y,";"`
Characters received:
`PA 10 20 80 90 ;`

(2) Instructions sent: `PRINT #1, "PA10,20,";X;",";Y;";"`
Characters received: `PA 10 20 80 90 ;`

The methods are similar, except for the use of commas versus semicolons between the parameters. The commas result in fields of 10 characters being sent to the plotter. If the parameters are less than 10 characters, this means unnecessary blanks are sent, resulting in extra activity over the interface bus. The semicolons compress these fields, except for leading and trailing blanks between the numeric variables.

Example — Plotting a Line Chart with Variable Parameters

The following BASIC program illustrates method (2) for sending variables as part of an HP-GL instruction. If you want to plot this line chart with different data points, simply substitute the new X,Y coordinate pairs in the DATA statements (lines 80 and 90).



ManualsPlus.com

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;SC-20,80,-10,60;"
30  FOR I=1 TO 7
40    READ X,Y
50    PRINT #1, "PA";X;",";Y;";PD;"
60  NEXT I
70  PRINT #1, "PU0,40,;PD0,0,60,0;SP0;"
80  DATA 0,10,10,12,20,18,30,22,40,17
90  DATA 50,25,60,24
100 END
```

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; assigns user-unit scale to the plotting area

30 BASIC statement for starting a loop

40 BASIC statement to read the X- and Y-coordinates from lines 80 and 90

50 moves to the point specified by the current X,Y coordinates — the first point is moved to with the pen up, and all subsequent points are moved to with the pen down

60 BASIC statement to close the loop

70 moves with the pen up to the top of the Y-axis; draws the Y- and X-axes with the pen down; returns the pen to the carousel

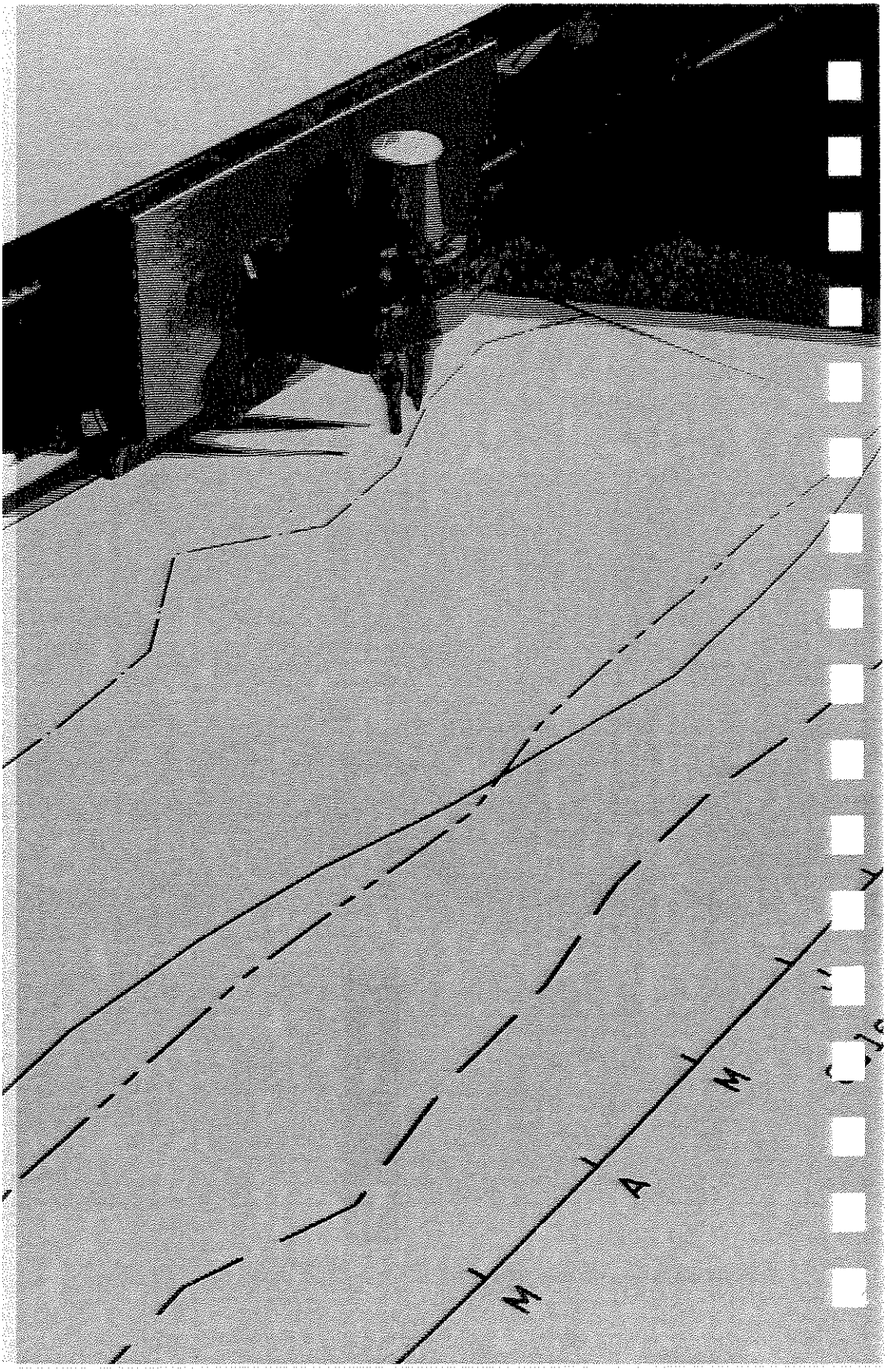
80 BASIC statement that declares the first five X,Y coordinate pairs to be read by line 40

90 BASIC statement that declares the next two X,Y coordinate pairs to be read by line 40

ManualsPlus.com

Notes

ManualsPlus.com



ManualsPlus.com

Chapter 5
Enhancing Your Plots

What You'll Learn in This Chapter

In this chapter you will learn how to: draw tick marks on axes, create grids, draw a symbol at each data point, and draw with dashed or dotted line patterns. Using these enhancements makes your data easier to interpret.

HP-GL Instructions Covered

XT The X-Tick Instruction
YT The Y-Tick Instruction
TL The Tick Length Instruction
SM The Symbol Mode Instruction
LT The Line Type Instruction

Terms You Should Understand

Tick — a small mark that indicates a certain number of units along an axis. Major ticks often mark every 5th or 10th unit, whereas minor ticks often mark single units.

Grid — a major tick that extends across the full height or width of a graph.

The Tick Instructions, XT and YT

USES: The XT instruction draws a vertical X-tick at the current pen location. The YT instruction draws a horizontal Y-tick at the current pen location. Use these instructions to draw tick marks on axes, and to draw grids.

SYNTAX: *XT term*
and
YT term

EXPLANATION: Both instructions cause a tick mark to be drawn at the current pen position; XT draws a vertical tick, and YT draws a

ManualsPlus.com

horizontal tick. They include an automatic pen down feature. After the tick is drawn, the current pen status (up or down) is restored.

You can vary the length of the tick mark with the tick length instruction, TL. However, if you do not specify a tick length, defaults are used as follows: 0.5% of $|P2_X - P1_X|$ for YT, and 0.5% of $|P2_Y - P1_Y|$ for XT.

The following table summarizes the possible error conditions or unexpected results that you might observe with the XT and YT instructions.

Condition	Error	Plotter Response
no parameters	none	draws tick
1 or more parameters	2	draws tick

Example — Drawing X-Ticks

The following example draws a horizontal line 3000 plotter units long, placing X-ticks 1000 plotter units apart. Refer to the TL instruction, described next, for another example that draws ticks.

```
"IN;SP2;PA200,500;XT;PD;PR1000,0;XT;"
"PR1000,0;XT;PR1000,0;XT;PU;SP0;"
```

The Tick Length Instruction, TL

USES: The TL instruction specifies the length of the tick marks on the X-axis and Y-axis. Use this instruction to set the lengths of both the positive and negative portions of tick marks, or to draw a grid.

SYNTAX: *TL* tp (, tn) *term*
or
TL *term*

Parameter	Format	Range	Default
tp and tn	decimal	0 to $2^{23}-1$ percentage	0.5% of $\|P2_X - P1_X\|$ and of $\|P2_Y - P1_Y\|$

EXPLANATION: A TL instruction without parameters (TL;) sets default values of 0.5,0.5. A TL instruction with only one parameter specifies the length of tp; tn is assumed to be 0 in this case. The parameters are interpreted as follows:

ManualsPlus.com

1. Tp. The positive tick length, tp, determines the length of the upward portion of the tick marks drawn along the X-axis, and the right-side portion of the tick marks drawn along the Y-axis.
2. Tn. The negative tick length, tn, determines the length of the downward portion of the tick marks drawn along the X-axis, and the left-side portion of the tick marks drawn along the Y-axis.

Both parameters are specified as a *percentage* of the vertical and horizontal distances between P1 and P2. Therefore, although a larger range is permitted, a more practical range is 0 to 100 percent. The parameter values are a percentage of the vertical scale length $|P2_Y - P1_Y|$ when used with the XT instruction. With the YT instruction, the parameter values are a percentage of the horizontal scale length $|P2_X - P1_X|$.

Note that the actual tick length is a function established by P1 and P2. Therefore, unless the area defined by P1 and P2 is square, the tick lengths on the X- and Y-axis will differ even if the same tick length percentage value is specified for both XT and YT.

Use the instruction with both parameters to draw a tick that extends on both sides of the axis. To suppress the negative portion of the tick, use the instruction with only one parameter. To suppress the positive portion of the tick, specify 0 (zero) for tp and specify the desired percentage for tn.

A TL instruction remains in effect until another TL instruction is executed or the plotter is initialized or set to default conditions.

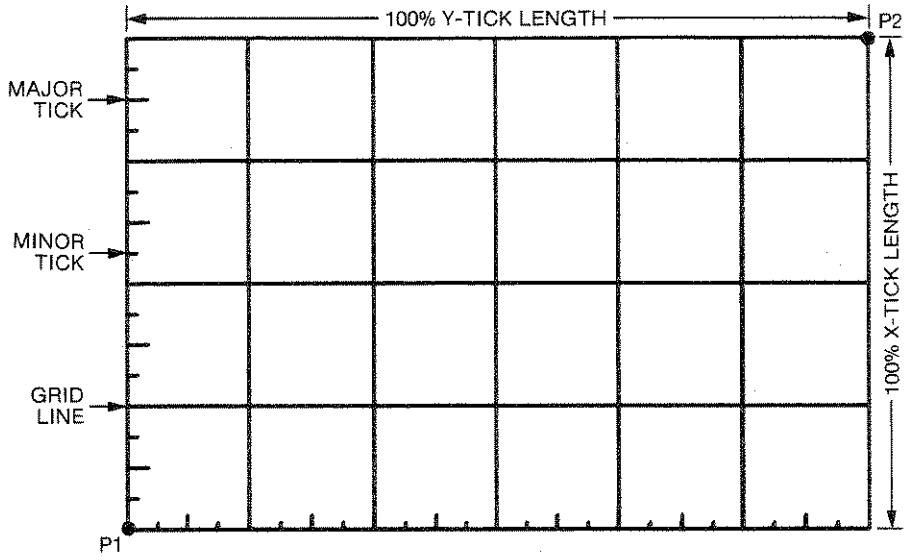
The following table summarizes the possible error conditions or unexpected results that you might observe with the TL instruction.

Condition	Error	Plotter Response
no parameters	none	sets tp and tn to 0.5%
1 parameter	none	draws only positive portion of tick when XT or YT is executed
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction

Example — Drawing Grid Lines, Major Ticks, and Minor Ticks

The following program illustrates how to draw grid lines, major ticks and minor ticks along the Y-axis and X-axis.

ManualsPlus.com




```
10  'Insert configuration statement here
20  PRINT #1, "IN;IP200,1000,4400,3800;"
30  PRINT #1, "SC0,60,0,40;SP2;PR0,40;EA60,0;"
40  FOR I=1 TO 4
50    PRINT #1, "PD;PR0,-2.5;TL1.5;YT;PR0,-2.5;"
60    PRINT #1, "TL3;YT;PR0,-2.5;TL1.5;YT;"
70    PRINT #1, "PR0,-2.5;TL100;YT;"
80  NEXT I
90  FOR J=1 TO 6
100   PRINT #1, "TL100;XT;PR2.5,0;TL1.5;XT;"
110   PRINT #1, "PR2.5,0;TL3;XT;PR2.5,0;"
120   PRINT #1, "TL1.5,0;XT;PR2.5,0;"
130 NEXT J
140 PRINT #1, "SP0;"
150 END
```

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; establishes scaling points P1 and P2

30 assigns user-unit scale; selects pen 2; sets the starting pen position; draws a rectangle around the P1/P2 area (the EA instruction is described in Chapter 6)

40 BASIC statement for starting a loop to repeat lines 50–70 four times in order to draw horizontal ticks and grids in the entire P1/P2 area

ManualsPlus.com

50	pen down; draws to first tick position; sets tick length of 1.5%; draws minor Y-tick; draws to next tick position
60	new tick length of 3%; draws major Y-tick; draws to next tick position; new tick length of 1.5%; draws minor Y-tick
70	draws to next tick position; new tick length of 100%; draws the grid line
80	BASIC statement to close the loop
90	BASIC statement to start a loop to repeat lines 100–120 six times to draw vertical ticks and grids in entire P1/P2 area
100–120	similar to lines 50–70, except that X-ticks are drawn
130	BASIC statement to close the loop
140	returns the pen to the carousel

The Symbol Mode Instruction, SM

USES: The SM instruction is used with the PA and PR instructions to draw a symbol at each X,Y coordinate. Use symbol mode plotting to draw a specified character at each data point to differentiate data contained in scattergrams, geometric drawings, or multiple-line graphs.

SYNTAX: *SM* character *term*
or
SM *term*

Parameter	Format	Range	Default
character	label	most printing characters (decimal codes 33–58 and 60–126)*	none

*The semicolon (decimal code 59) is an HP-GL terminator and cannot be used as a symbol *in any character set.* It is used only to cancel symbol mode.

EXPLANATION: After an SM instruction has been executed, subsequent PA and PR instructions function as described in Chapter 4, except that the symbol mode character is drawn at each X,Y coordinate, centered over the point. The SM instruction includes an automatic pen down feature. After the symbol is drawn, the current pen status (up or down) is restored.

The character is drawn in the character set selected at the time the SM instruction is executed. The character does not change even if a new character set is subsequently selected. (If the downloadable character set, defined by the DL instruction, is selected, the character is not centered over the points unless it was defined that way. Refer to

ManualsPlus.com

Chapter 11.) In addition, the size (SI and SR), slant (SL), and direction (DI and DR) instructions affect how the character is drawn. (Refer to Chapter 7.)

An SM instruction remains in effect until another valid SM instruction is executed or the plotter is initialized or set to default conditions.

The following table summarizes the possible error conditions or unexpected results that you might observe with the SM instruction.

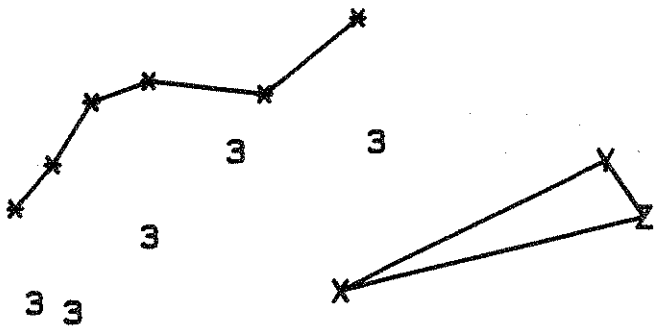
Condition	Error	Plotter Response
no parameter	none	cancels symbol mode
parameter out-of-range	3	ignores instruction*

*If the out-of-range parameter is a carriage return (decimal code 13), an error is not generated and symbol mode is canceled.

Example — Plotting in Symbol Mode

The following instructions cause symbols to be plotted in three situations: with the pen down for a line graph, with the pen up for a scattergram, and with the pen down for a geometric drawing.

```
"IN;SP1;SM*;PA200,1000;"
"PD400,1230,600,1560,900,1670,1500,1600,2000,2000;"
"PU;SM;PA100,300;SM3;"
"PA300,500,500,450,900,850,1350,1300,2100,1350;PU;"
"SM;PA1900,560;PD;SMY;PA3300,1250;"
"SMZ;PA3500,950;SMX;PA1900,560;PU;SP0;"
```



The Line Type Instruction, LT

USES: The LT instruction specifies the line pattern that will be used with AA, AR, CI, PA, PR, RA, RR, WG, FP, EP, FT, and UF instructions. Use the LT instruction so you can differentiate between curves on multiple-line graphs. Also use LT to emphasize or deemphasize other plotted lines and shapes.

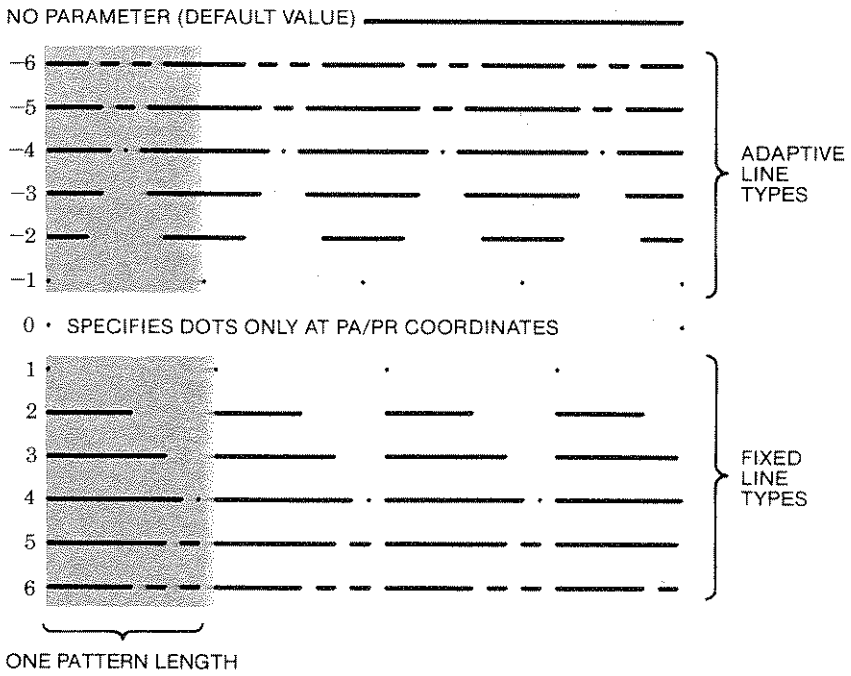
ManualsPlus.com

SYNTAX: *LT* pattern number (, pattern length) *term*
or
LT *term*

Parameter	Format	Range	Default
pattern number	integer	−6 to +6	no parameter (solid line)
pattern length	decimal	0 to $2^{23} - 1$	4% of the diagonal distance between P1 and P2

EXPLANATION: An LT instruction without parameters (LT;) defaults the line type to a solid line. The pattern number and pattern length parameters are interpreted as follows.

1. Pattern number. The pattern numbers and the line patterns they each generate are shown below. A positive pattern number uses the actual specified pattern length to draw lines. Any portion of a pattern that is not used to draw the specified line is carried over into the next line. A negative pattern number, on the other hand, automatically adjusts the specified pattern length so that each line contains one or more complete pattern segments. The positive patterns are known as "fixed line types," whereas the negative patterns are known as "adaptive line types."



ManualsPlus.com

2. Pattern length. The optional pattern length parameter specifies the length of one complete pattern and is expressed as a percentage of the diagonal distance between the scaling points P1 and P2. Therefore, although a larger range is permitted, a more practical range is 0 to 100 percent. If you do not specify a pattern length, the plotter uses the last value issued. If no pattern length has ever been specified, the plotter uses a length of 4%.

After an LT instruction is executed, subsequent vectors drawn with AA, AR, CI, PA, PR, EP, FP, WG, RA, and RR instructions will be drawn in the specified line type. The UF and FT instructions also use the current line type for designing area-fill patterns. (PA and PR are described in Chapter 4. The other instructions are described in Chapter 6). Once selected, the line type remains in effect until another valid LT instruction is executed or the plotter is initialized or set to default conditions.

NOTE: If you issue an arc (AA, AR) or circle (CI) instruction while a negative pattern number is in effect, the pattern length is adjusted to the chord length of the arc or circle. Because the chord length is usually very short, patterns cannot be easily distinguished (they often appear to be solid lines). Therefore, use positive pattern numbers when drawing arcs and circles. (Refer to the instruction descriptions in Chapter 6.)

You should also be aware that if a line segment ends in the pen up portion of a positive line pattern, a pen down instruction, PD, will not physically lower the pen until the next vector instruction is executed. Also, the pen up instruction, PU, clears any carry-over portion of a pattern segment; in other words, the next line will be drawn from the beginning of the line pattern. ■

The following table summarizes the possible error conditions or unexpected results that you might observe with the LT instruction.

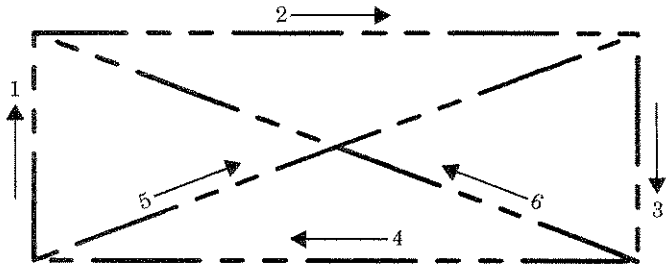
Condition	Error	Plotter Response
no parameters	none	establishes solid line type
1 parameter	none	uses the previously established pattern length
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction

ManualsPlus.com

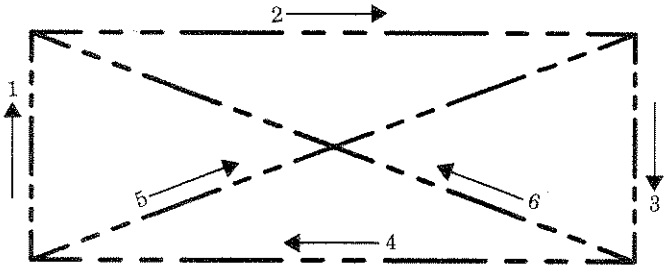
Example — Fixed and Adaptive Line Types

The following example demonstrates the difference between positive and negative pattern numbers. The program plots two rectangular boxes, one with line type 6 and one with line type −6. The pattern length in both cases is 20%. Notice that each line drawn with the adaptive pattern contains complete pattern segments, whereas the lines drawn with the fixed pattern contain partial pattern segments. In this case, unused portions of pattern segments are carried over into the next line. The lines are numbered in the order in which they are drawn.

```
"IN;IP0,0,4000,4000;"
"SC0,10,0,10;SP1;"
"LT6,20;PU1,6;PD1,9,9,9,9,6,1,6,9,9;"
"PU9,6;PD1,9;PU;"
"LT-6,20;PA1,1;PD1,4,9,4,9,1,1,1,9,4;"
"PU9,1;PD1,4;"
"SP0;"
```

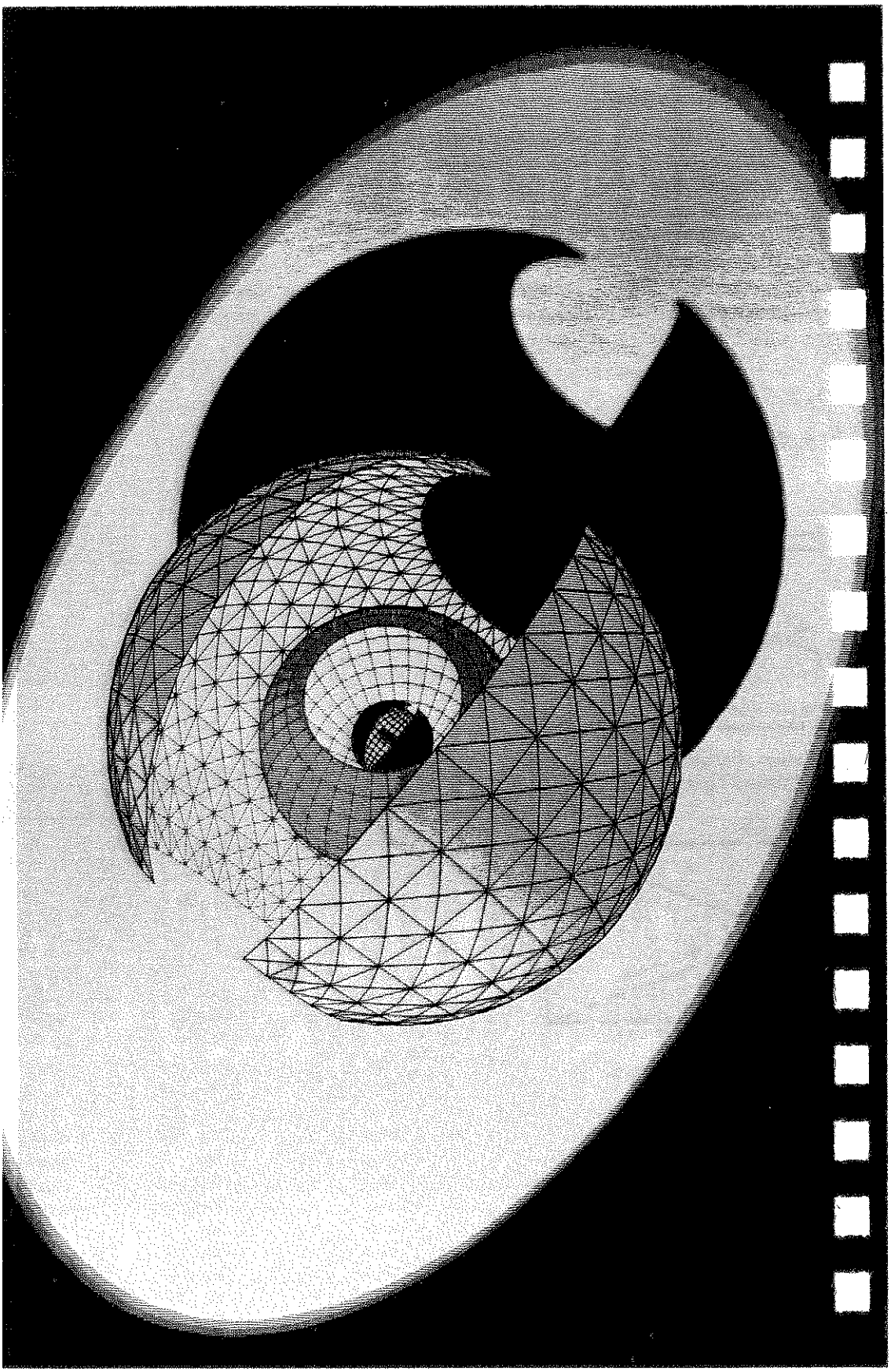


Fixed Pattern Length
(Line Type = 6; Pattern Length = 20)



Adaptive Pattern Length
(Line Type = −6; Pattern Length = 20)

ManualsPlus.com



ManualsPlus.com

Chapter 6
Circles, Arcs, and Polygons

What You'll Learn in This Chapter

In this chapter you will learn about plotting circles, arcs, and polygons. You can create some simple polygons (such as circles, arcs, and rectangles) using a single instruction. In addition, you can define your own polygons using the polygon mode instruction.

Besides creating polygons, you will also learn how to outline them and how to fill them with various shading patterns. These shading patterns include patterns offered by the plotter and patterns that you can design.

Finally, you will learn about the polygon buffer and how to determine the proper memory allocation for this buffer so that it will hold your polygon.

HP-GL Instructions Covered

CT The Chord Tolerance Instruction
CI The Circle Instruction
AA The Arc Absolute Instruction
AR The Arc Relative Instruction
FT The Fill Type Instruction
UF The User-Defined Fill Type Instruction
PT The Pen Thickness Instruction
WG The Fill Wedge Instruction
EW The Edge Wedge Instruction
RA The Fill Rectangle Absolute Instruction
EA The Edge Rectangle Absolute Instruction
RR The Fill Rectangle Relative Instruction
ER The Edge Rectangle Relative Instruction
PM The Polygon Mode Instruction
EP The Edge Polygon Instruction
FP The Fill Polygon Instruction

ManualsPlus.com

Terms You Should Understand

Arc — a portion of the circumference of a circle.

Wedge — the combination of an arc and the radii connecting the arc endpoints to the circle center. Wedge is equivalent to the geometric term "sector."

Chord — a straight line joining two points on an arc or the circumference of a circle.

Chord Tolerance — the allowable deviation from a perfectly smooth circle or arc. A perfectly smooth circle or arc is composed of an infinite number of chords; the human eye cannot discern the individual chords. However, the plotter cannot draw an infinite number of chords. Therefore, you need to specify an acceptable chord tolerance. The chord tolerance determines the number of chords (and thus the smoothness) for a circle or arc. The chord tolerance is defined in the CT instruction to be an angle in degrees, or a deviation distance in current units.

Polygon — any shape constructed of a number of points in such a way that lines connecting the points meet to form a closed area. Simple shapes such as circles, rectangles, and wedges are polygons. More complex shapes such as block letters can also be polygons. In this chapter, polygon most often refers to the shapes that are defined in polygon mode.

Subpolygon — a polygon defined as part of a larger polygon. For example, the block letter **O** is a polygon that consists of two subpolygons: the outside circle and the inside circle.

Vertex — a coordinate point that is used to define part of the shape of a polygon. To define a rectangle, for example, you would specify the coordinate points for each corner of the rectangle. These points are also called vertices.

Polygon Mode — a mode established by the PM instruction, used for defining polygons. In this mode, certain HP-GL instructions are collected in the polygon buffer (rather than being executed) so that they can be used to define the vertices of the polygon.

Polygon Buffer — a portion of the plotter's "graphics memory" that is used to store the vertices of the polygon that is currently being defined. The polygon can be one defined by the WG, EW, RA, EA, RR, ER, or PM instructions.

Edge — the outline of a polygon.

Fill Type — the shading pattern used in a polygon.

ManualsPlus.com

Hatch — a fill type that consists of parallel lines.

Cross-hatch — a fill type that consists of one set of parallel lines drawn at a 90-degree angle to another set of parallel lines.

The Chord Tolerance Instruction, CT

USES: The CT instruction establishes whether the chord tolerance parameter of all subsequent CI, AA, AR, WG, and EG instructions is interpreted as an angle in degrees or as a deviation distance in current units. (The chord tolerance parameter governs the smoothness of circles, arcs and wedges, as described later in the following explanation.) Since these instructions assume degrees mode, you only need to execute CT if you want to change to deviation-distance mode, or if you have previously specified deviation-distance mode but want to return to degrees mode.

SYNTAX: *CT* n *term*
or
CT *term*

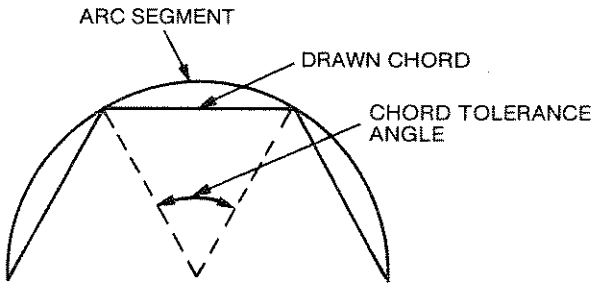
Parameter	Format	Range	Default
n	integer	0 or 1	0

EXPLANATION: A CT instruction without parameters (CT;) is equivalent to a CT instruction with the parameter 0, which sets degrees mode. The two possible parameter values are listed next.

NOTE: The CT instruction determines the mode for the chord tolerance parameter of subsequent CI, AA, AR, WG, and EW instructions. For simplicity, the following discussion refers to "circles." However, the discussion applies equally to circles, arcs, and wedges. ■

1. n = 0. A parameter of 0 sets degrees mode for subsequent chord tolerance parameters. When in degrees mode, the chord tolerance parameter sets the maximum angle at the end of whose radii a chord will be drawn to represent an arc segment. (The radii will not be drawn. Also, the actual angle may be slightly smaller than specified in order for the plotter to divide the circle into equal angles so that all chords are the same length.) The degrees mode is illustrated on the next page.

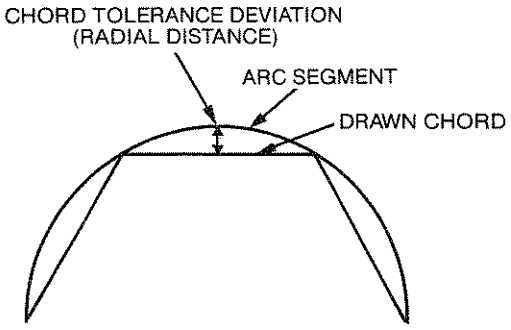
ManualsPlus.com



Degrees Mode

Note that specifying larger angles for the chord tolerance parameter, while maintaining the same radius of the circle, will cause fewer chords to be drawn. This decreases the smoothness of the circle. (Refer to the example under The Circle Instruction, CI.) Also note that increasing the radius of a circle, while maintaining the same angle, will result in longer chords (the *number* of chords remains the same, but the chords increase in *length* because the circle is larger). If the circle is large enough, this could make it appear less smooth.

2. n = 1. A parameter of 1 sets deviation-distance mode for subsequent chord tolerance parameters. When in deviation-distance mode, the chord tolerance parameter sets the maximum radial distance permitted between the chord drawn and the arc segment that it represents. This distance is expressed in current units. (The actual radial distance may be slightly less than specified in order for the plotter to divide the circle into chords of equal length.) The deviation-distance mode is illustrated below.



Deviation Distance Mode

Note that specifying a larger deviation distance for the chord tolerance parameter, while maintaining the same radius for the circle, will cause fewer chords to be drawn. This decreases the smoothness of the circle. Also note that increasing the radius of a circle, while maintaining the same deviation distance, will result in more chords

ManualsPlus.com

of a shorter length. The smoothness remains the same, but the circle takes longer to plot when it has more chords.

The current mode (degrees or deviation distance) remains in effect until another valid CT instruction is executed, or the plotter is initialized or set to default conditions.

The following table summarizes the possible error conditions or unexpected results that you might observe with the CT instruction.

Condition	Error	Plotter Response
no parameter	none	establishes degrees mode
more than 1 parameter	2	executes first parameter
parameter not 0 or 1	3	ignores instruction

The Circle Instruction, CI

USES: The CI instruction draws a circle of a specified radius and chord tolerance.

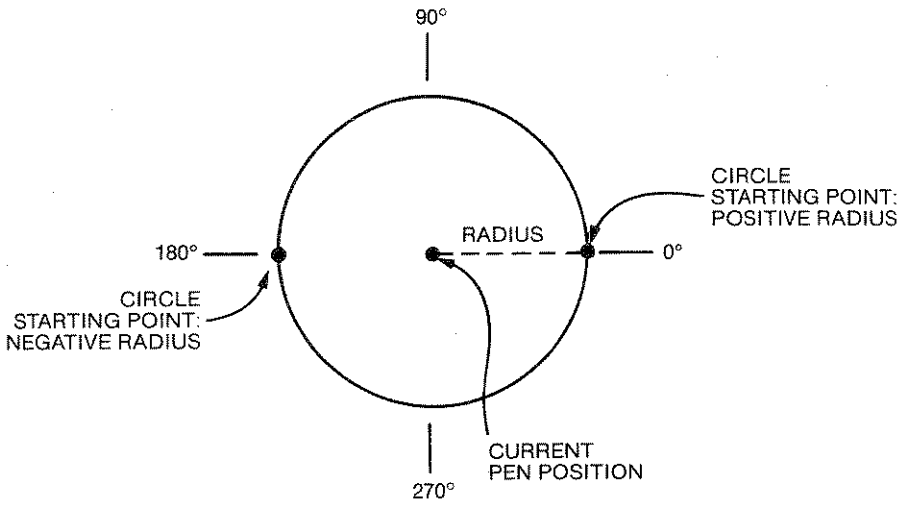
SYNTAX: *CI* radius (, chord tolerance) *term*

Parameter	Format	Range	Default
radius	decimal	-2^{23} to $2^{23}-1$ current units	none
chord tolerance	decimal	-2^{23} to $2^{23}-1$ current mode	5 degrees

EXPLANATION: The radius and chord tolerance parameters are interpreted as described in the following paragraphs and illustrations.

1. Radius. The radius determines the size of the circle. As shown on the next page, the sign of the radius parameter defines the starting point of the circle: a circle with a positive radius starts at the 0-degree point; a circle with a negative radius starts at the 180-degree point. The current pen position is the center of the circle. If scaling is off, the radius is in plotter units. If scaling is on, the radius is in user units. If user units are not the same size in the X- and Y-directions, ellipses will be drawn. (Refer to The Scale Instruction, SC, in Chapter 3.)

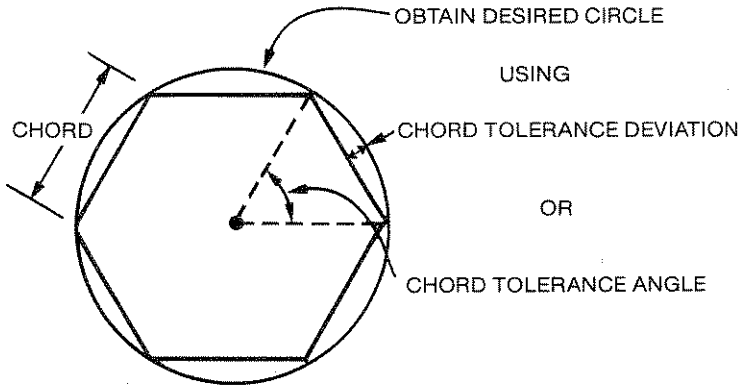
ManualsPlus.com



2. Chord Tolerance. A plotted circle is actually made up of a series of straight line segments, or chords, as shown below. Increasing the number of chords (by decreasing the chord tolerance parameter) increases the smoothness of the circle. However, this also increases the length of time required to draw the circle.

   The chord tolerance parameter is interpreted as either degrees or a deviation distance, depending on whether degrees mode or deviation mode has been set (by the CT instruction or default conditions). Refer to The Chord Tolerance Instruction, CT, earlier in this chapter. If degrees mode is set, the chord tolerance parameter is interpreted as modulo 360.

   The sign of this parameter is ignored. The default chord tolerance is 5 degrees, which causes the circle to be made up of 72 chords. The first example later in this section illustrates the effects of different chord tolerance angles.



ManualsPlus.com

The CI instruction includes an automatic pen down feature. When a CI instruction is received, the pen lifts (if it was down), moves from the center of the circle (the current pen position) to the starting point on the circumference, lowers the pen, draws the circle, then returns with the pen up to the center of the circle. After the circle is drawn, the current pen status (up or down) is restored. To avoid drawing lines to the center of the circle, move to and from the circle's center with the pen up.

Each chord of the circle is drawn using the currently defined line type. (Refer to The Line Type Instruction, LT, in Chapter 5.) Since each chord is small, patterns adjusted to chord endpoints are hard to distinguish. Therefore, use positive parameters in the LT instruction when drawing circles.

The following table summarizes the possible error conditions or unexpected results that you might observe with the CI instruction.

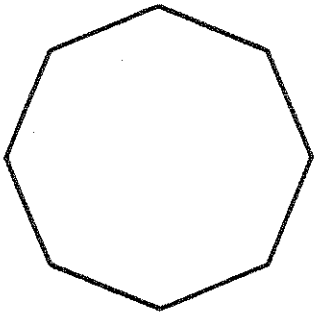
Condition	Error	Plotter Response
no parameters	none	ignores instruction
1 parameter	none	draws a circle with the specified radius and a chord tolerance of 5 degrees
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction
lost mode	none	ignores instruction (refer to Relationship of Plotting Instructions and Graphics Limits in Chapter 4)

Example — Effects of Chord Tolerance on Circle Smoothness

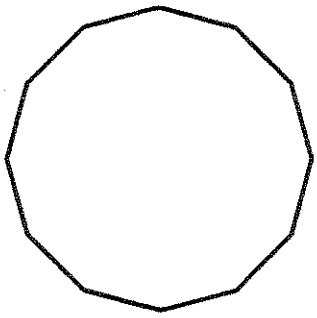
The following instructions draw circles with the same radius, but with different chord tolerance angles. Note that the circle becomes smoother as the chord tolerance parameter decreases. The circles are shown on the next page.

```
"IN;SP1;"
"PA3700,6050;CI800,45;"
"PA6300,6050;CI800,30;"
"PA3700,3950;CI800,15;"
"PA6300,3950;CI800,5;"
"SP0;"
```

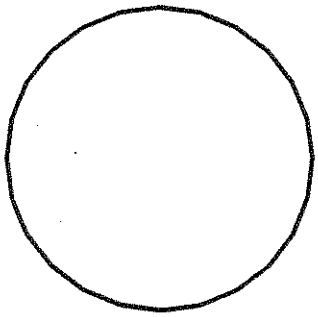
ManualsPlus.com



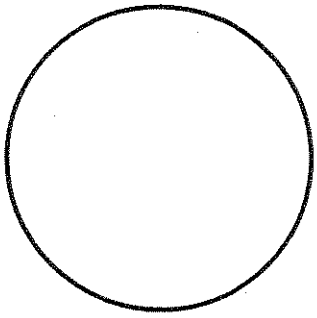
45-Degree Chord Angle



30-Degree Chord Angle



15-Degree Chord Angle



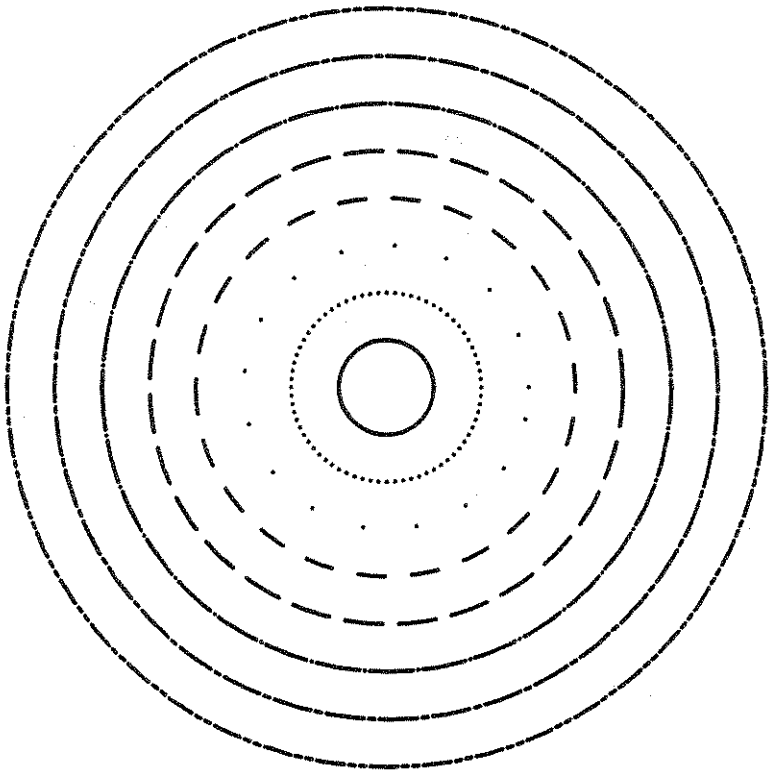
5-Degree Chord Angle

Example — Drawing Circles with Different Radii and Line Types

The following instructions draw eight circles with increasing radii and different line types. Positive (fixed) line types are specified. If negative (adaptive) line types had been specified, the line patterns would be indistinguishable; they would appear almost solid. This example includes the SC instruction to establish user units. Note that the user units are equal in the X- and Y-directions.

```
"IN;SP1;IP1000,1000,6000,6000;"
"SC-100,100,-100,100;"
"PR0,0;LT;CI10;LT0;CI-20;LT1;CI30;"
"LT2;CI-40;LT3;CI50;LT4;CI-60;"
"LT5;CI70;LT6;CI80;"
"SP0;"
```

ManualsPlus.com



The Arc Instructions, AA and AR

USES: The AA and AR instructions each draw an arc based on the present pen position and the specified center point. Use the arc absolute instruction, AA, to specify the center point in absolute coordinates. Use the arc relative instruction, AR, to specify the center point in relative coordinates.

SYNTAX: *AA* X-coordinate, Y-coordinate, arc angle (, chord tolerance) *term*

AR X-increment, Y-increment, arc angle (, chord tolerance) *term*

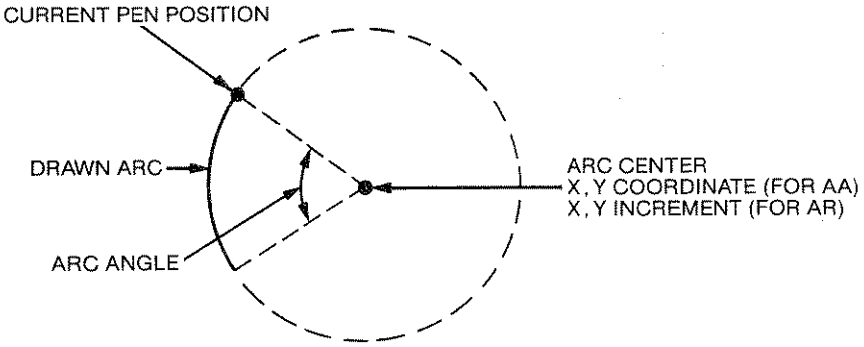
ManualsPlus.com

Parameter	Format	Range	Default
X- and Y-coordinates or X- and Y-increments	decimal	-2^{23} to $2^{23}-1$ current units	none
arc angle	decimal	-2^{23} to $2^{23}-1$ degrees	none
chord tolerance	decimal	-2^{23} to $2^{23}-1$ current mode	5 degrees

EXPLANATION: The AA and AR instructions are identical except for the way the center point is specified. For in-depth discussions of absolute and relative plotting, refer to Chapter 2 and to The Plot Absolute Instruction, PA, and The Plot Relative Instruction, PR, in Chapter 4.

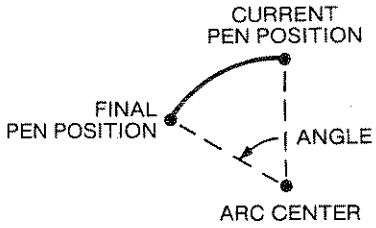
The parameters are interpreted as described in the following paragraphs and illustrations.

1. X,Y Coordinates (for AA) or X,Y Increments (for AR). These parameters specify the center of the arc in plotter units if scaling is off, or in user units if scaling is on. If user units are not the same size in the X- and Y-directions, distorted arcs will be drawn. The center of the arc is the center of the circle that would be drawn if the arc were 360 degrees. Refer to the following illustration.

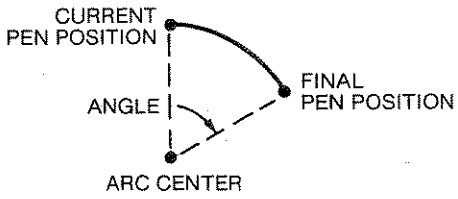


2. Arc Angle. The arc angle is the angle through which the arc is drawn. A positive angle draws counterclockwise from the current pen position, and a negative angle draws clockwise, as shown on the following page.

ManualsPlus.com



Positive Angle



Negative Angle

3. Chord Tolerance. The chord tolerance parameter is interpreted as either degrees or a deviation distance, depending on whether degrees mode or deviation mode has been set (by the CT instruction or default conditions). Refer to The Chord Tolerance Instruction, CT, earlier in this chapter.

   The sign of this parameter is ignored. The default chord tolerance is 5 degrees. As discussed under the CI and CT instructions, increasing the number of chords (by decreasing the chord tolerance parameter) generates a smoother arc.

Arcs are drawn starting at the current pen position using the current pen status (up or down) and line type. As with the CI instruction, you will find that positive parameters in the LT instruction produce more distinguishable line patterns than negative parameters produce. After the arc has been drawn, the pen position will remain at the end of the arc, rather than returning to the beginning.

The arc center can be located on or off the plotting surface. The plotter draws the arc according to the definitions given for plotting in Relationship of Plotting Instructions and Graphics Limits in Chapter 4. Remember that AR moves add to the present pen position. If cumulative moves result in a pen position that exceeds the range of -2^{23} to $2^{23}-1$, the plotter enters the lost mode.

The following table summarizes the possible error conditions or unexpected results that you might observe with the AA and AR instructions.

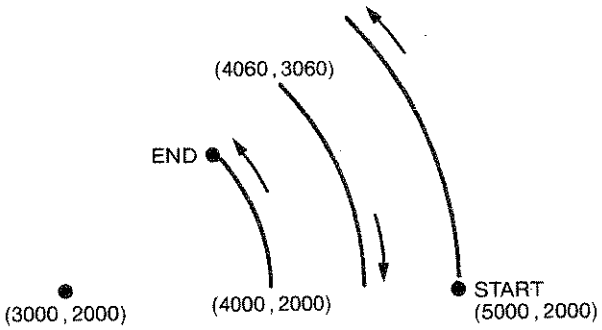
Condition	Error	Plotter Response
no parameters	none	ignores instruction
1 or 2 parameters	2	ignores instruction
more than 4 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction
lost mode	none	ignores instruction (refer to Relationships of Plotting Instructions and Graphics Limits in Chapter 4)

ManualsPlus.com

Example — Using the AA Instruction

The following instructions illustrate the effects of changing the distance to the center point while keeping the same arc angle. They also show the effect of changing the sign of the angle. Note that the pen remains down at the end of each arc, so you must lift the pen with the PU instruction in order to move to the beginning of the next arc without drawing a line.

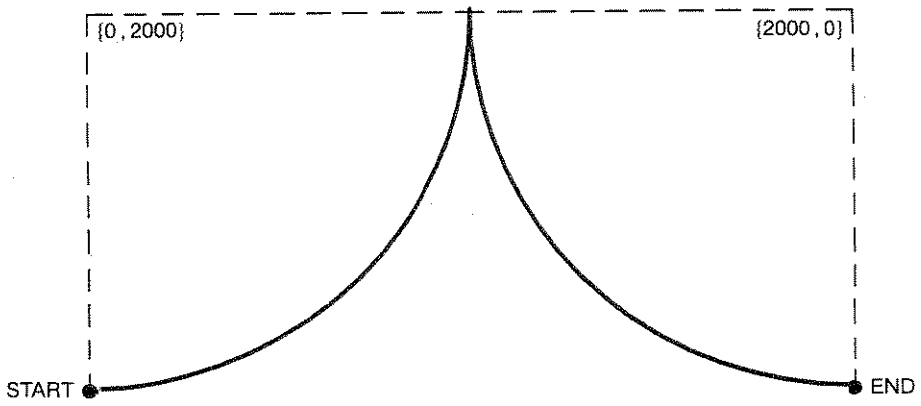
```
"IN;SP1;"
"PA 5000,2000;"
"PD;AA 3000,2000,45;"
"PU4060,3060;PD;AA 3000,2000,-45;"
"PU4000,2000;PD;AA 3000,2000,45;SP0;"
```



Example — Using the AR Instruction

The following instructions draw an arc centered around 0,2000 plotter units relative to the starting pen position, followed by an arc centered around 2000,0 plotter units relative to the new starting pen position. Note that the PD instruction is required to draw the arcs.

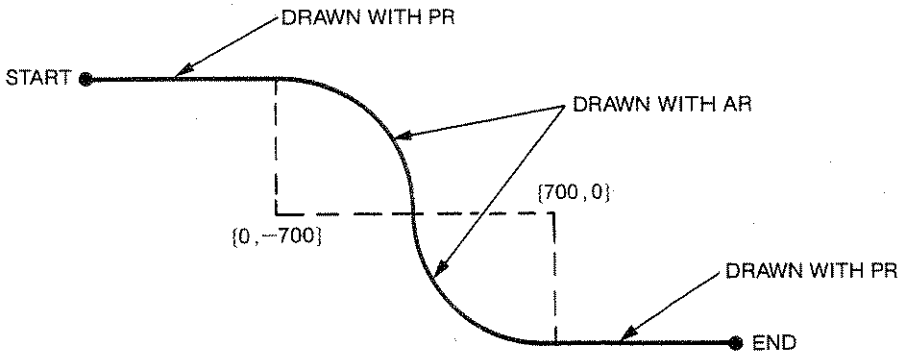
```
"IN;SP1;"
"PA10,10;PD;AR0,2000,90;AR2000,0,90;SP0;"
```



ManualsPlus.com

The next instructions show the effects of changing both the relative centers of the arcs, and the signs of the angles. The first arc is drawn clockwise around the center (negative angle), whereas the second arc is drawn counterclockwise (positive angle).

```
"IN;SP1;"
"PA10,5000;PD;PR1000,0;AR0,-700,-90;"
"AR700,0,90;PR1000,0;"
"SP0;"
```



The Fill Type Instruction, FT

USES: The FT instruction selects the type of area shading for use with an FP, RA, RR, or WG instruction. You can use this instruction to enhance pie charts, bar charts, and other charts with solid fill, parallel lines, cross-hatching, or a fill type of your own design.

SYNTAX: *FT* type (, spacing (, angle)) *term*
or
FT *term*

Parameter	Format	Range	Default
fill type	integer	1–6	1
spacing	decimal	0 to $2^{23}-1$ current units	depends on fill type
angle	decimal	-2^{23} to $2^{23}-1$ degrees, modulo 360	0 degrees

EXPLANATION: The FT instruction can have three parameters: fill type, spacing, and angle. If you omit parameters and this is the first FT instruction since the plotter was turned on or set to default conditions, the plotter uses the default parameter values. If this is not the first FT

ManualsPlus.com

instruction, and you omit parameters, the plotter uses the parameter values from the previous FT instruction. If you omit all parameters (FT;), the plotter uses the default values.

The following paragraphs list further details of each parameter.

1. Fill Type. The six parameters and corresponding fill types are listed below. "Bidirectional" means the pen draws back and forth; "unidirectional" means the pen draws in one direction only. For the highest quality solid fill on transparency film, use a unidirectional fill (type 2 or 6).

    1 solid bidirectional filling (lines with spacing as defined in the pen thickness instruction, PT, discussed later in this chapter)

    2 solid unidirectional filling (lines with spacing as defined in the PT instruction)

    3 parallel lines (always bidirectional for solid line types; unidirectional for dotted or dashed line types)

    4 cross-hatch (always bidirectional for solid line types; unidirectional for dotted or dashed line types)

    5 user-defined bidirectional (as defined by the user-defined fill type instruction, UF, described later in this chapter)

    6 user-defined unidirectional (as defined by the UF instruction)

    All fill types are drawn using the current pen. Fill types 3–6 use the current line type, whereas solid-fill types 1 and 2 always use a solid line. In this case, any line type specified by the LT instruction is ignored until the solid fill is completed.

    To use parameters 5 and 6, you must first define a fill style using the UF instruction. If you specify parameter 5 or 6 but do not define a fill style with UF, the FT instruction will be executed with the default UF (solid fill). The angle and the direction of fill will remain the same as specified in the FT instruction.

2. Spacing. The spacing parameter is ignored for solid-fill types 1 and 2; in these cases, spacing is determined by the PT instruction. For fill types 3 and 4, the spacing parameter is the distance between parallel lines in the fill area. For fill types 5 and 6, the spacing parameter is used by the UF instruction as the pattern repeat length.

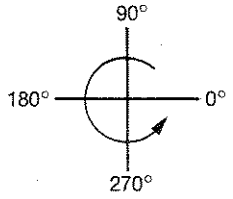
    For fill types 3 through 6, the spacing parameter is interpreted as plotter units if scaling is off or user units if scaling is on. If scaling is on and the units on the X- and Y-axes are not equal in size, the spacing parameter is interpreted using the units along the X-axis.

    The default is 1% of the diagonal distance between P1 and P2. Subsequent changes in the P1/P2 locations affect this distance and therefore

ManualsPlus.com

affect spacing. A parameter of 0 causes the plotter to use this default spacing.

3. Angle. The angle is referenced counterclockwise from the positive direction of the X-axis as shown below (0 and 180 are horizontal; 90 and 270 are vertical). The angle applies to all fill types. For cross-hatching, the first set of lines is drawn at the specified angle. The cross-hatched lines are then drawn at that angle plus 90 degrees.



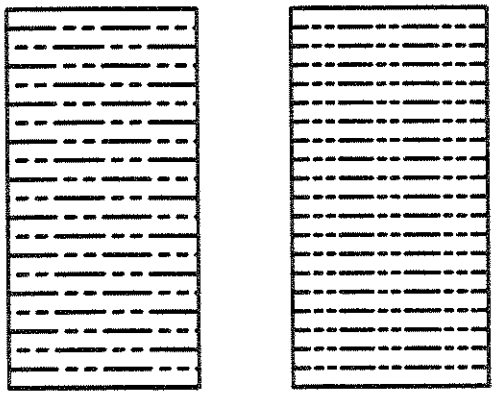
The following table summarizes the possible error conditions or unexpected results that you might observe with the FT instruction.

Condition	Error	Plotter Response
no parameters	none	establishes solid fill at an angle of 0 degrees
1 parameter	none	establishes specified fill type with spacing and angle of previous FT instruction
2 parameters	none	establishes specified fill type and spacing with angle of previous FT instruction
more than 3 parameters	2	executes first 3 parameters
parameter out-of-range	3	uses parameter of previous FT instruction

Example — Effects of Line Type Patterns

The following BASIC program shows the effects of positive and negative line types in filling a rectangular area. The positive line type produces an alternating pattern that can often be quite attractive. For more examples of area fill, refer to the following instructions, later in this chapter: The Wedge Instructions, WG and EW; The Absolute Rectangle Instructions, RA and EA; The Relative Rectangle Instructions, RR and ER; The Polygon Mode Instruction, PM; and the Fill Polygon Instruction, FP.

ManualsPlus.com




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA0,0;"
30  PRINT #1, "LT6;FT3,100,0;RA1000,2000;"
40  PRINT #1, "EA1000,2000;"
50  PRINT #1, "PA1500,0;"
60  PRINT #1, "LT-6;RA2500,2000;"
70  PRINT #1, "EA2500,2000;"
80  PRINT #1, "SP0;"
90  END
```

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; sets the starting pen position

30 selects line type 6 (fixed pattern); selects fill type 3 (parallel lines) with lines spaced every 100 plotter units at an angle of 0 degrees; defines and fills a rectangle (the RA and EA instructions are described later in this chapter)

40 edges the rectangle

50 sets the starting pen position for the next rectangle

60 selects line type −6 (adaptive pattern); defines and fills a rectangle using the same fill pattern specified in line 30

70 outlines the rectangle

80 returns the pen to the carousel

ManualsPlus.com

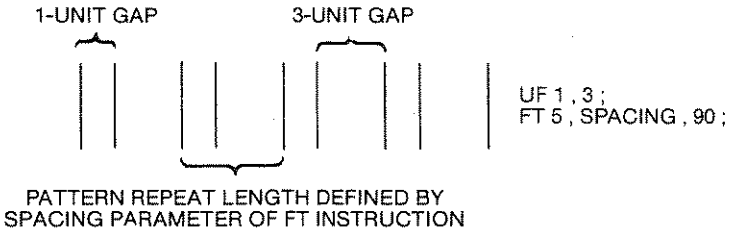
The User-Defined Fill Type Instruction, UF

USES: The UF instruction enables you to define a fill pattern composed of "gaps" between parallel lines. This is then the fill pattern used by parameters 5 and 6 in the FT instruction. Use this fill pattern with the RA, RR, WG, and FP instructions to generate unique line patterns such as a semilog pattern, or a candy-stripe effect.

SYNTAX: *UF* gap_1 (, gap_2, . . . gap_{20}) *term*
or
UF *term*

Parameter	Format	Range	Default
gap	integer	0 to $2^{23}-1$	none

EXPLANATION: The gap parameters for the UF instruction are unitless numbers that represent the distance to the next fill line. In order to access the fill style described by the UF instruction, you must also execute the FT instruction. You can use the parameters in the FT instruction to influence the appearance of your fill style. This is because the plotter fits the total number of gaps specified in the UF instruction into an area equal to the spacing parameter of the FT instruction. The spacing parameter thus becomes a pattern repeat length for the fill pattern you have defined. This concept is shown below, and further illustrated in the example at the end of this section.



NOTE: The pattern repeat length is always calculated with respect to the plotter-unit origin (0 , 0), regardless of where the shape you wish to fill begins. Thus, depending on where your shape begins in the plotting area, you might notice that your fill pattern does not always begin with the same gap parameter. ■

You can specify up to 20 parameters in one UF instruction. Each parameter must be a positive integer. You may include parameters equal to zero (0), but the sum of the parameters must be greater than zero. A UF instruction without parameters (UF;) produces solid fill.

ManualsPlus.com

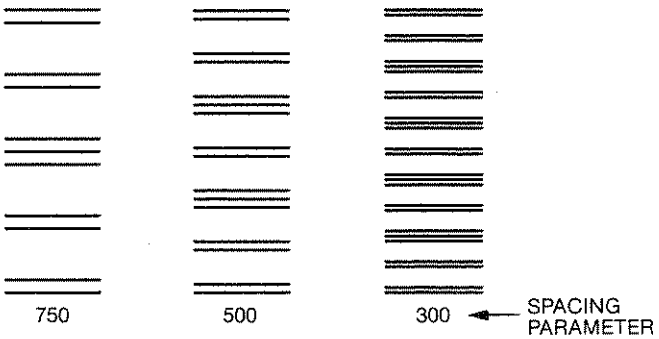
Lines are drawn using the current pen and line type. Because of the way gaps are defined, positive parameters in the LT instruction do not always produce the alternating fill pattern described under the FT instruction. Therefore, you will probably find that negative line-type parameters produce more satisfactory results. Lines are either bidirectional or unidirectional, depending on whether parameter 5 or 6 was specified in the FT instruction.

The following table summarizes the possible error conditions or unexpected results that you might observe with the UF instruction.

Condition	Error	Plotter Response
0 or 1 parameter	none	establishes solid fill
more than 20 parameters	2	executes first 20 parameters
parameter out-of-range or sum of gap parameters ≤ 0	3	ignores instruction

Examples — Creating Special Effects

The following BASIC program shows the effects of executing the same UF instruction, but decreasing the spacing parameter in the FT instruction to produce shorter pattern repeat lengths.




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;"
30  PRINT #1, "UF1,4,1,4,1;"
40  PRINT #1, "FT5,750,0;"
50  PRINT #1, "PA0,0;RA500,1500;"
60  PRINT #1, "FT5,500,0;"
70  PRINT #1, "PA1000,0;RA1500,1500;"
80  PRINT #1, "FT5,300,0;"
90  PRINT #1, "PA2000,0;RA2500,1500;"
100 PRINT #1, "SP0;"
110 END
```

ManualsPlus.com

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1

30 defines the fill pattern gaps

40 selects fill type 5 (unidirectional) with a spacing parameter of 750 plotter units at an angle of 0 degrees

50 sets the starting pen position; defines and fills a rectangle (the RA instruction is described later in this chapter)

60 changes the fill-type spacing parameter to 500 plotter units

70 sets the starting pen position for the next rectangle; defines and fills the rectangle

80 changes the fill-type spacing parameter to 300 plotter units

90 sets the starting pen position for the next rectangle; defines and fills the rectangle

100 returns the pen to the carousel

If the pattern repeat length is small, so that spaces between clusters of lines are much greater than the spaces between lines in a cluster, the appearance of different line widths can be created. The following program lines illustrate this effect.

```
"IN;SP1;PA3000,0;"
"UF 1,5,1,1,5,1,1,1,5,1,1,1,1,5;"
"FT 5,300,0;"
"RA 3500,1500;"
"SP0;"
```

You can also arrange gaps to produce a semilog pattern, as shown on the following page. (Hint: To be sure the pattern begins with the largest gap at the bottom, start the filled rectangle at Y = 0.)

ManualsPlus.com

```
"IN;SP1;PA4000,0;"
"UF 9,8,7,6,5,4,3,2,1;"
"FT 5,750,0;"
"RA 4500,1500;"
"SP0;"
```

The Pen Thickness Instruction, PT

USES: The PT instruction determines the optimum spacing between the lines drawn in a solid fill for rectangles, wedges, and polygons. Use this instruction with the FT, RA, RR, WG, and FP instructions to produce a good solid fill.

SYNTAX: *PT* pen thickness *term*
or
PT *term*

Parameter	Format	Range	Default
pen thickness	decimal	0.1 – 5.0 millimetres	0.3

EXPLANATION: The pen thickness parameter represents the physical pen-tip width in millimetres. The following table lists typical pen thicknesses. Specify the number that corresponds to the currently selected pen.

Pen Type	Thickness
Narrow fiber-tip (paper or transparency)	0.3*
Wide fiber-tip (transparency)	0.6*
Wide fiber-tip (paper)	0.7*
Roller-ball	0.3*
Drafting 4 × 0 (not commonly used)	0.18
Drafting 3 × 0 (not commonly used)	0.25
Drafting 0	0.35
Drafting 1	0.50
Drafting 2	0.70
Drafting 4	1.00

*The tip width is stamped on the end of each pen, along with P (for paper), T (for transparency), or R (for roller-ball).

ManualsPlus.com

The pen thickness determines (but is not identical to) the spacing of lines needed to produce a solid fill. The thicker the pen, the wider the gap allowed between lines. If your solid fill has gaps showing between the lines, adjust the pen thickness downward. If a fiber-tip pen is getting wider through wear, or if improved throughput is desired at the expense of less dense fill, adjust the pen thickness upward.

NOTE: Pen tip width is an average width. Some tips could be slightly narrower or wider. Adjustments might be necessary to compensate for these differences. ■

PT .7; FT 1 drawn with 0.7-mm wide pen. This example produces the optimum solid fill for a 0.7-mm wide pen.

PT .7; FT 1 drawn with 0.3-mm wide pen. This example plots faster than if PT .3 had been specified, but the solid fill is less dense.

The PT instruction pertains only to the currently selected pen. It remains in effect until:

- a new pen is selected using either an SP instruction or the front-panel controls
- a new PT instruction is executed
- the plotter is initialized or set to default conditions

The following table summarizes the possible error conditions or unexpected results that you might observe with the PT instruction.

Condition	Error	Plotter Response
more than 1 parameter	2	executes first parameter
parameter out-of-range	3	ignores instruction
new pen selected	none	establishes 0.3-mm pen thickness

The Wedge Instructions, WG and EW

USES: The fill wedge instruction, WG, fills any wedge (sector) of a circle; the edge wedge instruction, EW, outlines any wedge. Use these instructions to produce sectors of a pie chart.

ManualsPlus.com

SYNTAX: *WG* radius, start angle, sweep angle (, chord tolerance) *term*

EW radius, start angle, sweep angle (, chord tolerance) *term*

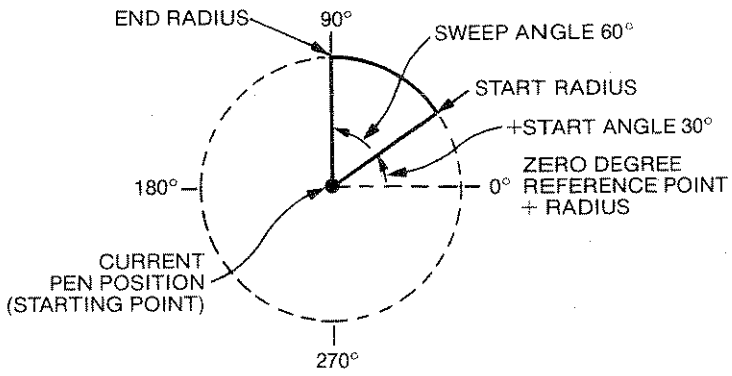
Parameter	Format	Range	Default
radius	decimal	-2^{23} to $2^{23}-1$ current units	none
start angle	decimal	-2^{23} to $2^{23}-1$ degrees, modulo 360	none
sweep angle	decimal	-2^{23} to $2^{23}-1$ degrees	none
chord tolerance	decimal	-2^{23} to $2^{23}-1$ current mode	5 degrees

EXPLANATION: The WG and EW instructions are interpreted by the plotter in the same way. The only difference is that the WG instruction produces a filled wedge, whereas the EW instruction produces an outlined wedge.

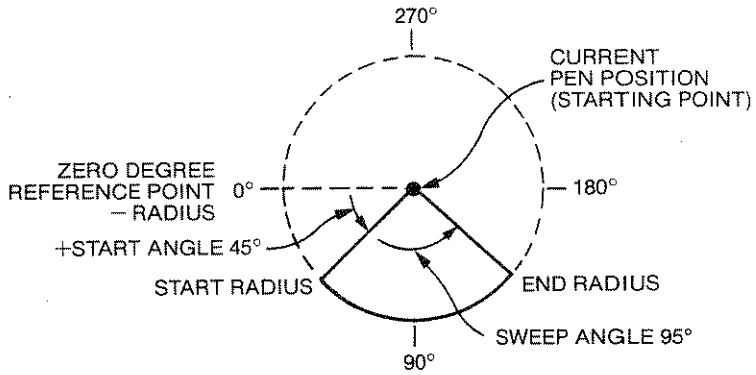
The first three parameters are required since they have no default values. The next paragraphs describe each parameter.

1. Radius. The radius specifies the distance from the current pen position to the start of the wedge's arc. The radius is in plotter units if scaling is off and user units if scaling is on. If user units are not the same size in the X- and Y-directions, distorted wedges will be drawn. The sign of the radius determines the zero-degree reference point for the start angle and sweep angle, as shown on the next page.
2. Start Angle. The start angle specifies where the radius is first drawn. A positive start angle positions the radius counterclockwise from the zero-degree reference point; a negative start angle positions the radius clockwise from the zero-degree reference point.
3. Sweep Angle. The sweep angle specifies the number of degrees through which the arc of the wedge is drawn from the start angle. Parameters greater than 360 are set to 360. A positive sweep angle draws the arc counterclockwise; a negative sweep angle draws the arc clockwise.

ManualsPlus.com



Positive Radius



Negative Radius

4. Chord Tolerance. The chord tolerance parameter is interpreted as either degrees or deviation distance, depending on whether degrees mode or deviation mode has been set (by the CT instruction or default conditions). Refer to The Chord Tolerance Instruction, CT, earlier in this chapter.

   The sign of this parameter is ignored. The default chord tolerance is 5 degrees. As discussed under the CI and CT instructions, increasing the number of chords (by decreasing the chord tolerance parameter) generates a smoother arc.

The WG instruction defines and fills a wedge using the current pen and fill type (and the current line type if the fill is not solid). The EW instruction defines and edges a wedge using the current pen and a solid line (regardless of the current line type defined by LT).

With both WG and EW, the starting point for the wedge is the current pen position, which should be thought of as the center of the circle that would be drawn if the wedge were 360 degrees. The WG and EW instructions include an automatic pen down feature. Upon completion

ManualsPlus.com

of the wedge, the pen returns to the current pen position and the pen status is restored.

NOTE: The WG and EW instructions clear the polygon buffer and then use this buffer for the wedge definition before filling or edging the wedge. In order for these instructions to work, you must be sure enough memory has been allocated in the polygon buffer to hold the number of chords in the wedge. The default memory is sufficient for even the largest wedge possible (125 chords). If you wish to change the polygon buffer size, refer to The Polygon Buffer at the end of this chapter. ■

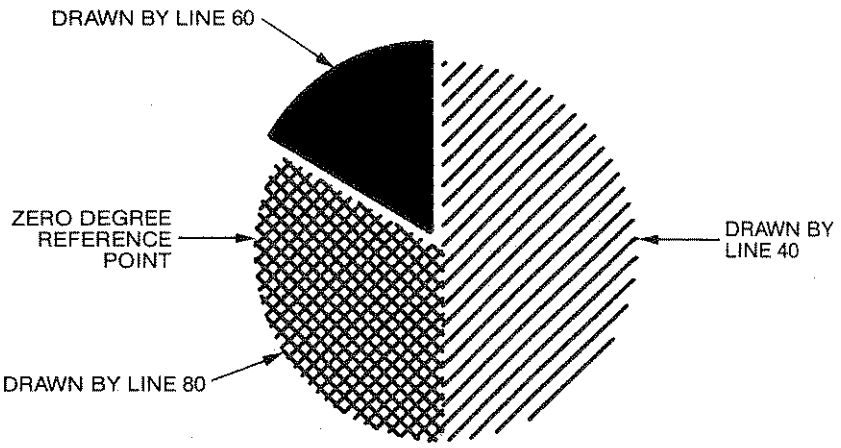
The following table summarizes the possible error conditions or unexpected results that you might observe with the WG and EW instructions.

Condition	Error	Plotter Response
no parameters	none	ignores instruction
fewer than 3 parameters	2	ignores instruction
more than 4 parameters	2	executes first 4 parameters
parameter out-of-range	3	ignores instruction
polygon buffer overflow	7	if WG, does not fill the wedge; if EW, edges whatever is in the buffer. All data that overflowed is lost.
lost mode	none	ignores instruction (refer to Relationship of Plotting Instructions and Graphics Limits in Chapter 4)

Example — Defining and Filling Wedges Using the WG Instruction

The following BASIC program illustrates how to use the WG instruction to creat a pie chart with one wedge offset for emphasis. Following this program is another example that shows two methods for outlining the wedges.

ManualsPlus.com




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP2;PA7000,2000;"
30  PRINT #1, "FT3,75,45;"
40  PRINT #1, "WG-1000,90,180;"
50  PRINT #1, "SP3;PR-60,110;FT1,0,0;"
60  PRINT #1, "WG-1000,270,60;"
70  PRINT #1, "SP4;PA7000,2000;FT4,60,45;"
80  PRINT #1, "WG-1000,330,120;"
90  PRINT #1, "SP0;"
100 END
```


Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 2; sets the starting pen position (center of the pie)

30 selects fill type 3 (parallel lines) with lines spaced every 75 plotter units at an angle of 45 degrees

40 defines and fills a wedge based on the zero-degree reference point for a negative radius of 1000 plotter units, starting at 90 degrees, sweeping counterclockwise through 180 degrees

50 selects pen 3; sets a new pen position which changes the pie's center point in order to offset the wedge; selects fill type 1 (solid, bidirectional at a 0-degree angle with spacing defined by default PT .3 established by IN in line 20)

60 defines and fills a wedge with the same zero-degree reference point and radius, starting at 270 degrees, sweeping counterclockwise through 60 degrees

ManualsPlus.com

70 selects pen 4; sets the pen position back to the original pie center; selects fill type 4 (cross-hatch) with lines spaced every 60 plotter units at an angle of 45 degrees

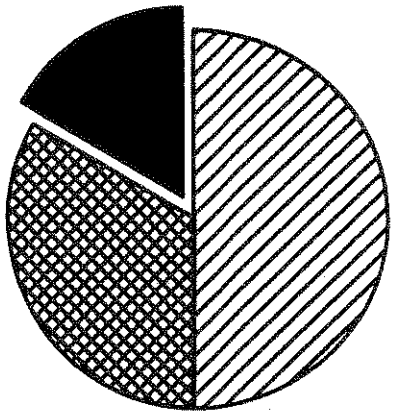
80 defines and fills a wedge with the same zero-degree reference point and radius, starting at 330 degrees, sweeping counterclockwise through 120 degrees

90 returns the pen to the carousel

Example — Outlining the Wedges Using the EW or EP Instructions

A good graphics programming habit is to outline filled areas *after* they have been filled. This creates a clean edge and minimizes ink bleed that could result if an area were outlined *before* being filled.

There are two methods for outlining a wedge. One uses the edge wedge instruction, EW, and the other uses the edge polygon instruction, EP. Both methods are presented below, along with reasons that you might use each method. Both methods produce the same plot, which is shown first.



Method A. To edge the wedges in the previous program using the EP instruction, simply add these lines to that program. (The complete program, including these three lines, is shown at the end of Chapter 1 under Developing a Plot.)

```
45 PRINT #1, "EP;"

65 PRINT #1, "EP;"

85 PRINT #1, "EP;"
```

Notice that an EP instruction follows each WG instruction. This is because EP outlines whatever currently resides in the polygon buffer. Only one wedge resides in the buffer at a

ManualsPlus.com

given time; therefore EP must be executed following each wedge. The EP instruction is very simple and efficient because it does not use parameters. Refer to The Edge Polygon Instruction, EP, later in this chapter.

Method B. To edge the wedges in the previous program using the EW instruction, simply add these lines to that program. (If you have already added lines 45, 65, and 85 in method A, replace them with the following lines.)

```
45 PRINT #1, "EW-1000,90,180;"

65 PRINT #1, "EW-1000,270,60;"

85 PRINT #1, "EW-1000,330,120;"
```

Notice that the EW instruction requires parameters to define the wedge before outlining it (unlike the EP instruction). This increases the chance for typographical errors when entering the instruction in the computer. However, the EW instruction is more flexible than the EP instruction, because you can place EW instructions anywhere in the program, before or after any WG instructions. You can also execute an EW instruction alone (without an accompanying WG instruction).

The Absolute Rectangle Instructions, RA and EA

USES: The fill rectangle absolute instruction, RA, defines and fills a rectangle; the edge rectangle absolute instruction, EA, defines and outlines a rectangle. Use these instructions to fill and/or outline rectangles using absolute coordinates. The RA and EA instructions make it easy to create any chart that requires many rectangles; for example, a bar chart, flow chart, or organization chart. (To fill and outline rectangles using relative coordinates, refer to The Relative Rectangle Instructions, RR and ER, next in this chapter.)

SYNTAX: *RA* X-coordinate, Y-coordinate *term*

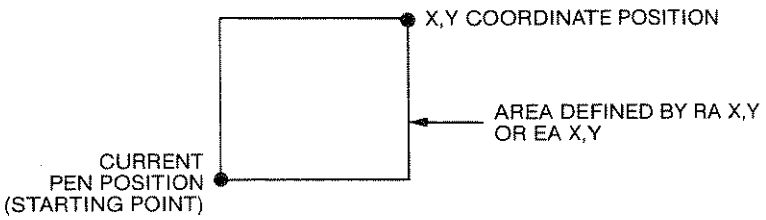
EA X-coordinate, Y-coordinate *term*

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none

EXPLANATION: The RA and EA instructions are interpreted by the plotter in the same way. The only difference is that the RA instruction produces a filled rectangle, whereas the EA instruction produces an outlined rectangle.

ManualsPlus.com

The current pen position is the starting point of the rectangle to be defined. The X,Y coordinates specify the opposite diagonal corner of the rectangle, as shown in the following illustration. Parameters are in plotter units if scaling is off and in user units if scaling is on.



NOTE: The illustration shows the current pen position in the lower-left corner and the X,Y parameters in the upper-right corner. However, both positions can be in any corner as long as the X,Y parameters are in a corner diagonally opposite to the current pen position. ■

The RA instruction defines and fills a rectangle using the current pen and fill type (and the current line type if the fill is not solid). The EA instruction defines and edges a rectangle using the current pen and a solid line (regardless of the current line type defined by LT.)

Both the RA and EA instructions include an automatic pen down feature. Upon completion of the rectangle, the plotter returns to the starting point and restores the pen status.

NOTE: The RA and EA instructions clear the polygon buffer and then use this buffer for the rectangle definition before filling or edging the rectangle. In order for these instructions to work, you must be sure enough memory has been allocated in the polygon buffer. A rectangle requires enough memory to hold five points. The default memory is sufficient; however, if you wish to change the polygon buffer size, refer to The Polygon Buffer at the end of this chapter. ■

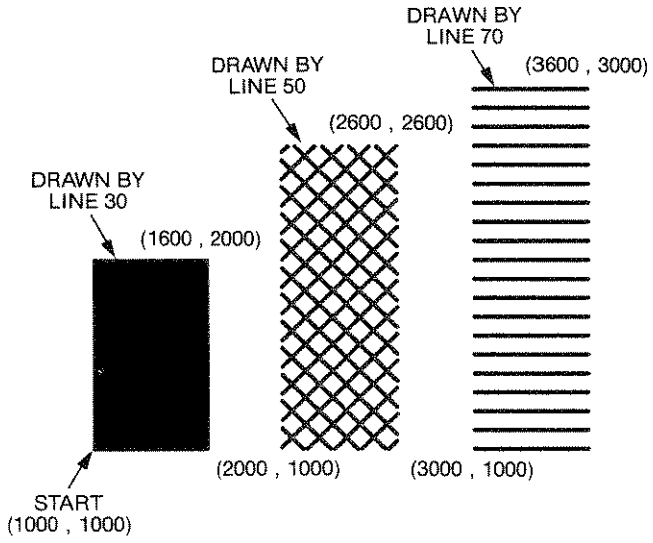
The table on the following page summarizes the possible error conditions or unexpected results that you might observe with the RA and EA instructions.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	ignores instruction
1 parameter	2	ignores instruction
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction
polygon buffer overflow	7	if RA, does not fill the rectangle; if EA, edges whatever is in the buffer. All data that overflowed is lost.
lost mode	none	ignores instruction (refer to Relationship of Plotting Instructions and Graphics Limits in Chapter 4)

Example — Defining and Filling Rectangles Using the RA Instruction

The following BASIC program demonstrates the use of the RA instruction with the FT instruction to create rectangles such as those you might use in a bar chart. (To create the same rectangles with relative coordinates, refer to The Relative Rectangle Instructions, RR and ER, next in this chapter.) Following this program is another example that shows two methods for outlining the rectangles.



ManualsPlus.com

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA1000,1000;"
30  PRINT #1, "FT1;RA1600,2000;"
40  PRINT #1, "PA2000,1000;"
50  PRINT #1, "FT4,100,45;RA2600,2600;"
60  PRINT #1, "PA3000,1000;"
70  PRINT #1, "FT3,100,0;RA3600,3000;"
80  PRINT #1, "SP0;"
90  END
```

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; sets the starting pen position for the first rectangle

30 selects fill type 1 (solid, bidirectional at an angle of 0 degrees); defines and fills the rectangle (with default PT .3 established by IN in line 20)

40 sets the starting pen position for the next rectangle

50 selects fill type 4 (cross-hatch) with lines spaced every 100 plotter units at an angle of 45 degrees; defines and fills the rectangle

60 sets the starting pen position for the next rectangle

70 selects fill type 3 (parallel lines) with lines spaced every 100 plotter units at an angle of 0 degrees; defines and fills the rectangle

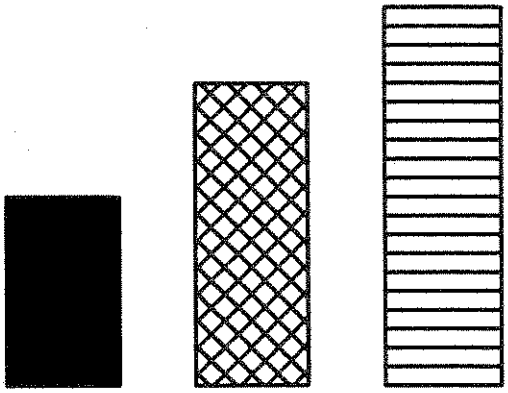
80 returns the pen to the carousel

Example — Outlining the Rectangles Using the EA or EP Instructions

A good graphics programming habit is to outline filled areas *after* they have been filled. This creates a clean edge and minimizes ink bleed that could result if an area were outlined *before* being filled.

There are two methods for outlining a rectangle. One uses the edge rectangle absolute instruction, EA, and the other uses the edge polygon instruction, EP. Both methods are presented on the next page, along with reasons that you might use each method. Both methods produce the same plot, which is shown first.

ManualsPlus.com



Method A. To edge the rectangles in the previous program using the EP instruction, simply add these lines to the program.

```
35 PRINT #1, "EP;"

55 PRINT #1, "EP;"

75 PRINT #1, "EP;"
```

Notice that an EP instruction follows each RA instruction. This is because EP outlines whatever currently resides in the polygon buffer. Only one rectangle resides in the buffer at a given time; therefore EP must be executed following each rectangle. The EP instruction is very simple and efficient because it does not use parameters. Refer to The Edge Polygon Instruction, EP, later in this chapter.

Method B. To edge the rectangles in the previous program using the EA instruction, simply add these lines to that program. (If you have already added lines 35, 55, and 75 in method A, replace them with the following lines.)

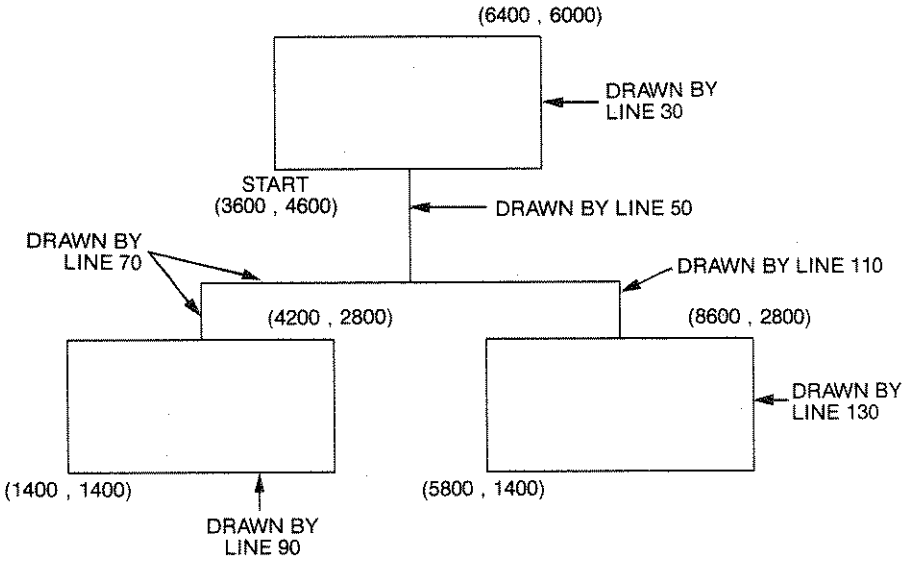
```
35 PRINT #1, "EA1600,2000;"

55 PRINT #1, "EA2600,2600;"

75 PRINT #1, "EA3600,3000;"
```

Notice that the EA instruction requires parameters to define the rectangle before outlining it (unlike the EP instruction). This increases the chance for typographical errors when entering the instruction in the computer. However, the EA instruction is more flexible than the EP instruction, because you can place EA instructions anywhere in the program, before or after any RA instructions. You can also execute an EA instruction alone (without an accompanying RA instruction), as shown in the following example.

ManualsPlus.com

Example — Drawing an Organization Chart with the EA Instruction

You can also outline rectangles for use in flow charts, organization charts, and scheduling charts. The following BASIC program illustrates how you can use the EA instruction to set up the outline of an organization chart.




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA3600,4600;"
30  PRINT #1, "EA6400,6000;"
40  PRINT #1, "PA5000,4600;"
50  PRINT #1, "PD5000,3400;"
60  PRINT #1, "PU7200,3400;"
70  PRINT #1, "PD2800,3400,2800,2800;"
80  PRINT #1, "PU4200,2800;"
90  PRINT #1, "EA1400,1400;"
100 PRINT #1, "PA7200,3400;"
110 PRINT #1, "PD7200,2800;"
120 PRINT #1, "PU8600,2800;"
130 PRINT #1, "EA5800,1400;"
140 PRINT #1, "SP0;"
150 END
```

ManualsPlus.com

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; sets the starting pen position using absolute coordinates

30 defines and outlines the top rectangle

40 moves to the starting pen position for the first vertical connecting line and establishes absolute coordinates for all subsequent PU and PD instructions

50 draws the first vertical connecting line

60 moves to the starting pen position for the horizontal connecting line

70 draws the horizontal connecting line, followed by the left vertical connecting line

80 moves to the starting pen position for the left rectangle

90 defines and outlines the left rectangle

100 moves to the starting pen position for the right vertical connecting line

110 draws the right vertical connecting line

120 moves to the starting pen position for the right rectangle

130 defines and outlines the right rectangle

140 returns the pen to the carousel

The Relative Rectangle Instructions, RR and ER

USES: The fill rectangle relative instruction, RR, defines and fills a rectangle; the edge rectangle relative instruction, ER, defines and outlines a rectangle. Use these instructions to fill and/or outline rectangles using relative coordinates. The RR and ER instructions make it easy to create any chart that requires many rectangles; for example, a bar chart, flow chart, or organization chart. (To fill and outline rectangles using absolute coordinates, refer to The Absolute Rectangle Instructions, RA and EA, earlier in this chapter.)

SYNTAX: *RR* X-increment, Y-increment *term*

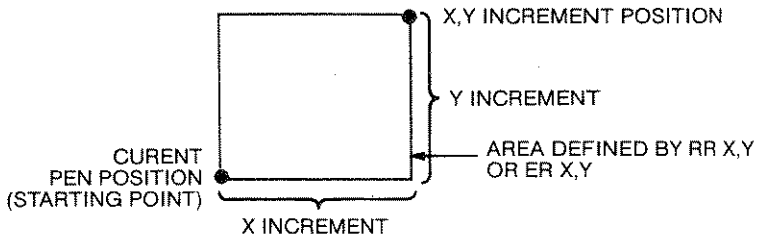
ER X-increment, Y-increment *term*

ManualsPlus.com

Parameter	Format	Range	Default
X- and Y-increments	decimal	-2^{23} to $2^{23}-1$ current units	none

EXPLANATION: Refer to The Absolute Rectangle Instructions, RA and EA, described previously, for complete details of how rectangle instructions are interpreted by the plotter. Unlike the EA instruction, the ER instruction outlines a rectangle using the current line type.

The main difference between the RA/EA instructions and the RR/ER instructions is that RA and EA use absolute coordinates to define a rectangle, whereas RR and ER use relative coordinates. The use of relative coordinates by the RR/ER instructions is shown next. For a complete discussion of the differences between absolute and relative coordinates, refer to Chapter 2 and to the PA and PR instructions in Chapter 4.



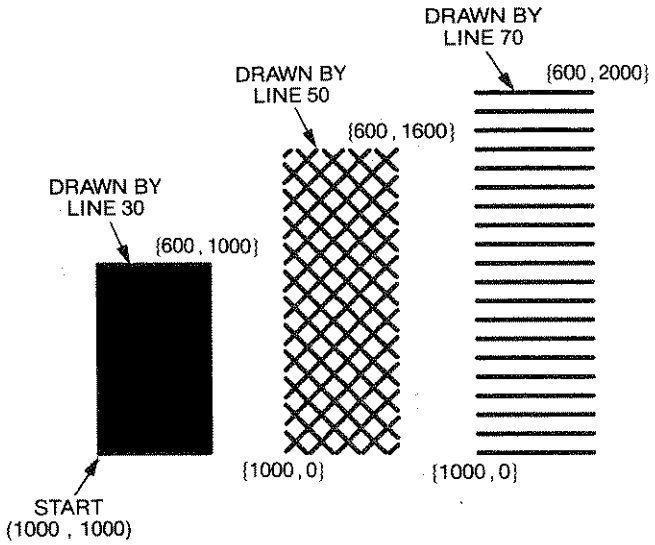
NOTE: The illustration shows the current pen position in the lower-left corner and the X,Y parameters in the upper-right corner. However, both positions can be in any corner as long as the X,Y parameters are in a corner diagonally opposite to the current pen position. ■

Examples — Using Relative Coordinates to Draw Rectangles

The following examples use the relative rectangle instructions to draw the same plots presented previously for the absolute rectangle instructions. Compare these programs with the previous programs to understand the differences between the coordinates used. Program explanations are not included here, because the explanations are the same as for the previous programs, except for the types of coordinates used (relative instead of absolute).

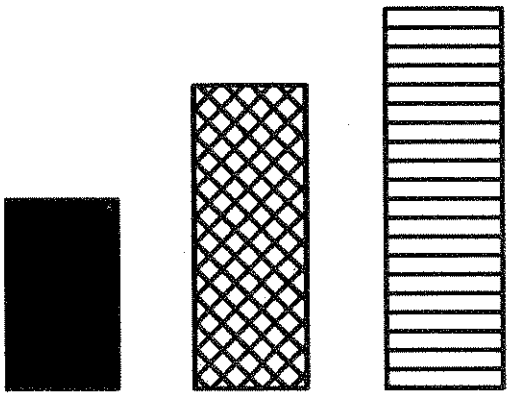
ManualsPlus.com

Defining and Filling Rectangles Using the RR Instruction




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PR1000,1000;"
30  PRINT #1, "FT1;RR600,1000;"
40  PRINT #1, "PR1000,0;"
50  PRINT #1, "FT4,100,45;RR600,1600;"
60  PRINT #1, "PR1000,0;"
70  PRINT #1, "FT3,100,0;RR600,2000;"
80  PRINT #1, "SP0;"
90  END
```

Outlining the Rectangles Using the ER or EP Instructions



ManualsPlus.com

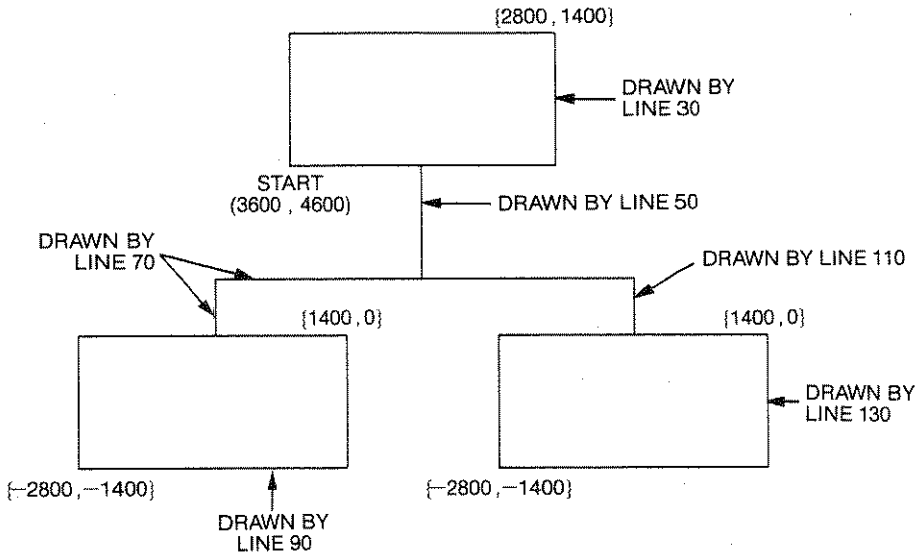
Method A (using EP). Add these lines to the previous program.

```
35  PRINT #1, "EP;"

55  PRINT #1, "EP;"

75  PRINT #1, "EP;"
```

Method B (using ER). Add these lines to the previous program.

```
35  PRINT #1, "ER600,1000;"

55  PRINT #1, "ER600,1600;"

75  PRINT #1, "ER600,2000;"
```

Drawing an Organization Chart with the ER Instruction




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA3600,4600;"
30  PRINT #1, "ER2800,1400;"
40  PRINT #1, "PR1400,0;"
50  PRINT #1, "PD0,-1200;"
60  PRINT #1, "PU2200,0;"
70  PRINT #1, "PD-4400,0,0,-600;"
80  PRINT #1, "PU1400,0;"
90  PRINT #1, "ER-2800,-1400;"
100 PRINT #1, "PR3000,600;"
110 PRINT #1, "PD0,-600;"
120 PRINT #1, "PU1400,0;"
130 PRINT #1, "ER-2800,-1400;"
140 PRINT #1, "SP0;"
150 END
```

ManualsPlus.com

The Polygon Mode Instruction, PM

USES: The PM instruction places the plotter in polygon definition mode. In this mode, you can construct polygons using other HP-GL instructions. These instructions are stored in the polygon buffer area of the plotter's graphics memory; they are not executed by the plotter until the polygon is drawn. The PM instruction only defines the polygon. In order to draw the polygon, you must fill it with the FP instruction and/or outline it with the EP instruction. Use the polygon instructions to design shapes such as block letters and logos.

SYNTAX: *PM* n *term*

Parameter	Format	Range	Default
n	integer	0–2	0

EXPLANATION: The PM instruction accepts only three parameters, as follows.

0 Clears the polygon buffer and enters polygon mode

1 Closes current subpolygon

2 Closes current subpolygon and exits polygon mode

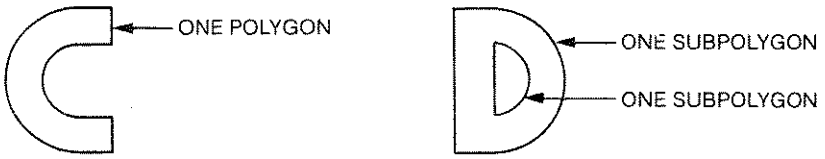
In addition, the plotter recognizes only the following HP-GL instructions while it is in polygon mode. (All other instructions are ignored.) The first column lists those instructions that are not executed by the plotter; instead, they are stored in the polygon buffer, where they can be accessed by the EP and FP instructions to draw the polygon. The second column lists the instructions that are executed by the plotter.

Stored in Polygon Buffer		Executed by Plotter	
PM	Polygon mode instruction	IN	Initialize instruction (exits polygon mode)
PA/PR	Plotting instructions	*	All output instructions
PU/PD	Pen instructions		
AA/AR	Arc instructions		
CI	Circle instruction		
CT	Chord tolerance instruction		

* OA, OC, OD, OE, OF, OG, OH, OI, OK, OL, OO, OP, OS, OT, OW

While in polygon mode, you can define either one polygon, or a series of subpolygons. For example, the block letter **C** is one complete polygon. However, the block letter **D** consists of two subpolygons: the outline and the "hole."

ManualsPlus.com



- To define one polygon (e.g., the letter **C**): move the pen to the desired starting location on your plot. Then execute PM 0 to enter polygon mode and specify the appropriate HP-GL instructions to define the shape of the polygon. Finally, execute PM 2 to exit polygon mode.
- To define a series of subpolygons (e.g., the letter **D**): move the pen to the desired starting location on your plot. Then execute PM 0 to enter polygon mode and specify the appropriate HP-GL instructions to define the shape of the first subpolygon; end this subpolygon with PM 1. Now specify the appropriate HP-GL instructions to define the shape of the next subpolygon, ending it with PM 1. (Continue in this way for any more subpolygons.) Finally, execute PM 2 to exit polygon mode.

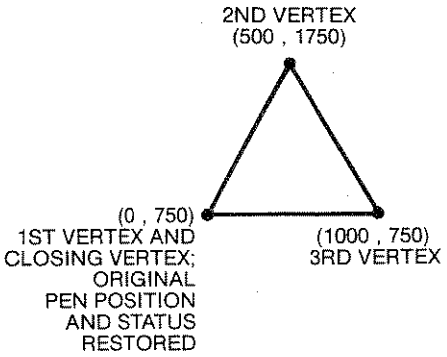
The current pen position *before* PM 0 is the first point (vertex) of the polygon, and thus the first point stored in the polygon buffer. Each subsequent point defines a vertex of the polygon. If you also include PM 1, the point *after* the PM 1 is the first point of the next subpolygon. This point is always moved to with the pen up, regardless of the current pen status. Each subsequent point defines a vertex of the subpolygon.

The vertices can be defined with the pen up or down. However, if you intend to outline the polygon with the EP instruction, note that EP only draws those points that are defined with the pen down. The FP instruction, on the other hand, fills all vertices, regardless of the pen up/down status.

It is good programming practice to "close" the polygon before executing PM 1 or PM 2, as shown in the following illustration. "Closing" a polygon means adding the final vertex that defines a continuous shape. However, if you have not explicitly closed the polygon, PM 1 and PM 2 do force the closure. In this case, executing PM 1 or PM 2 adds a vertex to close the polygon. The original pen position and status (up/down) are restored after PM 2 is executed.

ManualsPlus.com

```
"IN;SP1;PU0,750;PM0;"
"PD500,1750,1000,750,0,750;PM2;EP;"
```



NOTE: The PM instruction clears the polygon buffer and then uses the buffer to hold the polygon definition. In order for the PM instruction to work, you must allocate enough memory in the polygon buffer before you execute the PM instruction. The default buffer size is sufficient for a polygon with 127 points (vertices). For full details on determining the proper memory allocation, refer to The Polygon Buffer at the end of this chapter. ■

The following table summarizes the possible error conditions or unexpected results that you might observe with the PM instruction.

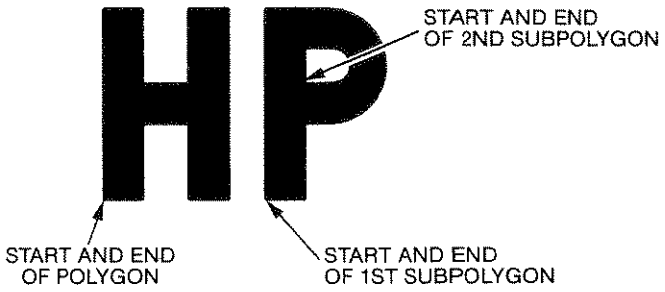
ManualsPlus.com

Condition	Error	Plotter Response
no parameter	none	clears the polygon buffer and enters polygon mode
parameter other than 0, 1, or 2	3	ignores instruction
PM 1 or PM 2 executed before PM 0	none	ignores PM 1 or PM 2 instruction (therefore, executes any HP-GL instructions after the PM 1 or PM 2, rather than placing them in the polygon buffer; and ignores subsequent FP and EP instructions)
use of any HP-GL instruction other than those listed in previous table, while in polygon mode	1	ignores illegal instruction
too many points; polygon buffer overflow	7	ignores all points that overflowed
polygon mode not exited (PM 2 not executed after PM 0)	none	ignores all instructions after PM 0
lost mode	none	ignores instruction (refer to Relationship of Plotting Instructions and Graphics Limits in Chapter 4)

Example — Creating Block Letters in Polygon Mode

The following BASIC program demonstrates how you can define the letter H as a polygon, and the letter P as two subpolygons. To design polygons, it is often helpful to draw them first on grid paper, in order to help you determine the proper coordinates. The letters in this program are defined using relative coordinates, starting at the lower-left corner of each letter and moving up and clockwise around the letter. Remember, the plotter will not draw the polygons until instructed to fill or edge them. For details, refer to The Fill Polygon Instruction, FP, and The Edge Polygon Instruction, EP, next in this chapter.

ManualsPlus.com




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA0,0;"
30  PRINT #1, "PM0;PR;PD0,1000,200,0,0,-400;"
40  PRINT #1, "PD250,0,0,400,200,0,0,-1000,-200,0;"
50  PRINT #1, "PD0,400,-250,0,0,-400,-200,0;"
60  PRINT #1, "PU;PM2;"
70  PRINT #1, "FP;EP;PU850,0;"
80  PRINT #1, "PM0;PD0,1000,325,0;"
90  PRINT #1, "AR0,-300,-180;"
100 PRINT #1, "PD-125,0,0,-400,-200,0;PM1;"
110 PRINT #1, "PU200,600;PD0,200,150,0;"
120 PRINT #1, "AR0,-100,-180;"
130 PRINT #1, "PD-150,0;PU;PM2;"
140 PRINT #1, "FP;EP;"
150 PRINT #1, "SP0;"
160 END
```


Program Explanation

NOTE: This program shows polygon mode definitions broken into readable segments of strings. For example, the definition for the polygon "H" is spread out over lines 30–60. You can combine these lines and send longer strings; the only limitation is the ability of your controller to output long strings. ■

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; sets the initial pen position

30 clears the polygon buffer and enters polygon mode; specifies relative coordinates; lowers the pen and defines three vertices of the letter H (beginning from the lower-left corner)

40 defines the next five vertices of the H

50 defines the last four vertices of the H (including the closing vertex at the lower-left corner)

60 lifts the pen (as a precaution against unwanted lines being drawn); exits polygon mode

ManualsPlus.com

70 fills the H using default values (solid, bidirectional fill); outlines the H; moves with the pen up to the starting position for the outside of the letter P

80 clears the polygon buffer and enters polygon mode; lowers the pen and defines two vertices of the outline of the P (beginning from the lower-left corner)

90 defines the outside rounded portion of the P, still using relative coordinates, and a counterclockwise angle of 180 degrees

100 defines the last three vertices of the outline of the P (including the closing vertex at the lower-left corner); closes this subpolygon

110 moves with the pen up to the starting position for the inside of the P; defines three vertices of the inside of the P

120 defines the inside rounded portion of the P

130 defines the final (closing) vertex of the inside of the P; lifts the pen; exits polygon mode

140 fills the P; outlines the P

150 returns the pen to the carousel

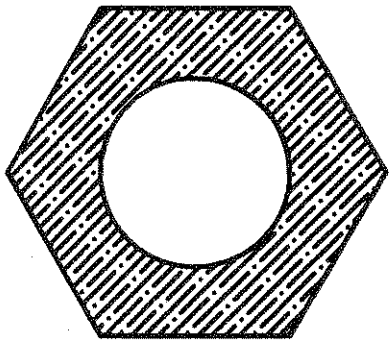
Example — Using the CI Instruction in Polygon Mode

A circle is interpreted slightly differently from the other HP-GL instructions that can be used in polygon mode. The difference is that a circle is always considered to be a complete subpolygon. That is, when the CI instruction is used in polygon mode, the plotter treats it as if it were preceded and followed by PM 1. The first coordinate points entered after a CI instruction thus become the first vertex of the next subpolygon. If a circle is to be the first element of a polygon, be sure to move the pen to the desired center point before executing PM 0.

As noted previously, PM 1 closes the current subpolygon if it is open. This usually adds another vertex to the subpolygon and changes the pen position. Thus, if you do not close the current subpolygon before executing a CI instruction, the pen position could change. Since the CI instruction is based on the current pen position, this means that the location of the circle could change. You should also note that there is no need to specify PD, as the CI instruction includes an automatic pen down feature. Remember also that when the circle is complete, the pen returns to the last position before the CI instruction was executed.

The following BASIC program demonstrates the use of the CI instruction in defining the two subpolygons needed to draw a hexagonal nut. Notice that the alternating line patterns are aligned even though there is a center "island."

ManualsPlus.com



```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA3000,3000;"
30  PRINT #1, "PM0;CI1000,60;CI500;PM2;"
40  PRINT #1, "LT4;FT3,50,45;FP;LT;EP;"
50  PRINT #1, "SP0;"
60  END
```


Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; sets the initial pen position (the center of the circle)

30 clears the polygon buffer and enters polygon mode; specifies a circle with a radius of 1000 plotter units and chord tolerance of 60 degrees as the first subpolygon (the outer hexagon); specifies a circle with a radius of 500 plotter units and default chord tolerance of 5 degrees as the second subpolygon (the inner circle); exits polygon mode

40 selects line type 4 (dashed lines); defines fill type 3 (parallel lines) with lines spaced every 50 plotter units at an angle of 45 degrees; fills the polygon; selects a solid line type; edges the polygon

50 returns the pen to the carousel

The Edge Polygon Instruction, EP

USES: The EP instruction outlines the polygon that is currently stored in the polygon buffer. Use this instruction to edge polygons that have been defined with the PM, RA, RR, and WG instructions.

SYNTAX: *EP term*

EXPLANATION: The EP instruction outlines any polygon that has been previously placed in the polygon buffer. Valid polygons include those defined by the PM, RA, RR, and WG instructions. While the EP instruction does access the data in the polygon buffer, note that EP does not clear the buffer or change the data in any way.

ManualsPlus.com

Only vertices that were defined with the pen down are edged. These are edged using the current pen and line type. Upon completion of the EP instruction, the original pen position and status (up/down) are restored.

For examples using the EP instruction, refer to the RA, RR, WG, and PM instructions in this chapter.

The following table summarizes the possible error conditions or unexpected results that you might observe with the EP instruction.

Condition	Error	Plotter Response
previous PM, RA, RR, or WG instruction overflowed the polygon buffer	none	edges only those points that remain in the buffer
no polygon previously defined	none	ignores instruction
1 or more parameters	2	edges the currently stored polygon

The Fill Polygon Instruction, FP

USES: The FP instruction fills the polygon that is currently stored in the polygon buffer, using the fill type specified in the FT instruction. Use the FP instruction to fill polygons that have been defined with the PM, EA, ER, and EW instructions.

SYNTAX: *FP term*

EXPLANATION: The FP instruction fills the polygon that has been previously placed in the polygon buffer. Valid polygons include those defined by the PM, EA, ER, and EW instructions. While the FP instruction does access the data in the polygon buffer, note that FP does not clear the buffer or change the data in any way.

The polygon is filled using the current pen, fill type, and line type. If subpolygons are defined, the FP instruction fills alternating areas, beginning with the outside area. (To see how this works, refer to the examples at the end of this section.) Upon completion of the FP instruction, the original pen position and status (up/down) are restored.

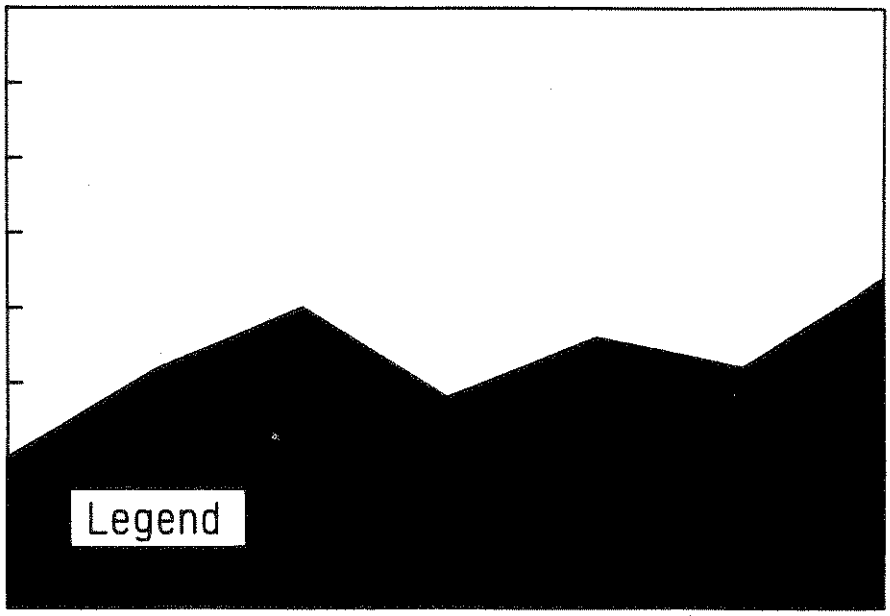
The following table summarizes the possible error conditions or unexpected results that you might observe with the FP instruction.

Condition	Error	Plotter Response
previous PM, RA, RR, or WG instruction overflowed the polygon buffer	none	ignores instruction; does not fill polygon
no polygon previously defined	none	ignores instruction
1 or more parameters	2	fills the currently stored polygon

ManualsPlus.com

Example — Creating a Surface Chart

The following BASIC program shows how to use the PM and FP instructions to draw and fill a curve in a surface chart, leaving a blank area for the legend. (This program also uses labeling and tick instructions that are described in Chapters 7 and 5, respectively.)




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SC-20,80,-10,60;"
30  PRINT #1, "SP2;PA0,0;"
40  PRINT #1, "PM0;PD0,10,10,16;"
50  PRINT #1, "PD20,20,30,14,40,18,50,16;"
60  PRINT #1, "PD60,22,60,0,0,0;PM1;"
70  PRINT #1, "PU4,4;PD4,8,16,8,16,4,4,4;PU;PM2;"
80  PRINT #1, "FP;EP;"
90  PRINT #1, "SP1;SI.25,.5;"
100 PRINT #1, "PU5.5,5;LBLegend"+CHR$(3)
110 PRINT #1, "PA0,0;ER60,40;"
120 PRINT #1, "TL1.5,0;"
130 FOR X=0 TO 60 STEP 10
140   PRINT #1, "PA";X;",0;XT;"
150 NEXT X
160 PRINT #1, "TL1,0;"
170 FOR Y=0 TO 40 STEP 5
180   PRINT #1, "PA0,";Y;";YT;"
190 NEXT Y
200 PRINT #1, "SP0;"
210 END
```

ManualsPlus.com

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; scales the plotting area to user units (subsequent plotting instructions will now be interpreted as user units)

30 selects pen 2; sets the initial pen position

40 clears the polygon buffer and enters polygon mode; lowers the pen and defines two vertices of the outside of the surface curve (beginning from the user-unit origin)

50 defines the next four vertices

60 defines the last three vertices (including the closing vertex at 0 , 0); closes this subpolygon

70 moves with the pen up to the starting position for the rectangular "legend" area (this is also the first vertex); lowers the pen and defines four vertices (including the closing vertex); lifts the pen; exits polygon mode

80 fills the surface curve using default fill conditions (solid, bidirectional with default PT .3 established by IN in line 20); edges the surface curve and legend area

90 selects pen 1; defines the character size

100 moves with the pen up to the starting position for the label; draws the label ("Legend") — refer to The Label Instruction, LB, in Chapter 7 for an explanation of the label terminator CHR$(3)

110 moves to the user-unit origin; defines and edges a rectangle around the chart

120 defines the size of the X-axis tick marks

130 BASIC statement that starts the loop for drawing the X-axis ticks at intervals of 10 user units

140 draws to the tick location; draws the tick

150 BASIC statement that ends the loop

160 defines the size of the Y-axis tick marks

170 BASIC statement that starts the loop for drawing the Y-axis ticks at intervals of 5 user units

180 draws to the tick location; draws the tick

190 BASIC statement that ends the loop

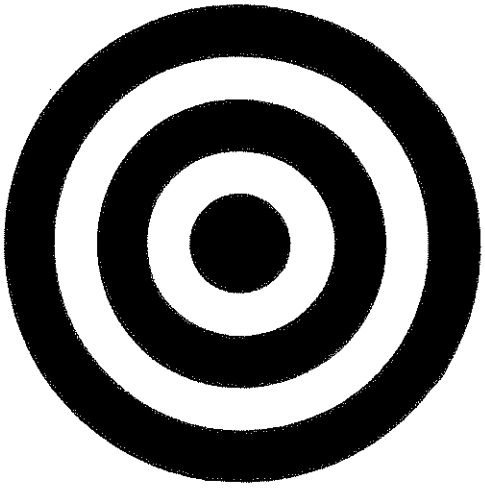
200 returns the pen to the carousel

ManualsPlus.com

Example — Filling Alternating Subpolygons

The following BASIC program shows the effects of defining and filling several subpolygons within one PM 0 instruction. In addition, it shows how to use the ESC.T instruction to increase the polygon buffer size in order to accommodate all of the subpolygons. (For further details, refer to The Polygon Buffer at the end of this chapter, and The Allocate Configurable Memory Size Instruction, ESC.T in Chapter 14.)

If you are using the RS-232-C interface, you can probably use the program example as is. To use this program as written, your computer must be able to read output statements; otherwise, replace lines 20 through 40 with this line: `20 PRINT #1, "GM5110;"`



```
10  'Insert configuration statement here
20  PRINT #1, CHR$(27)+".T;5110:"
30  PRINT #1, CHR$(27)+".L"
40  INPUT #1, L
50  PRINT #1, "IN;SP1;PA5000,5000;"
60  PRINT #1, "PM0;CI250;CI500;CI750;"
70  PRINT #1, "CI1000;CI1250;PM2;"
80  PRINT #1, "FP;EP;"
90  PRINT #1, "SP0;"
100 END
```

NOTE: The HP Touchscreen (150) computer **cannot** read output statements when using the HP-IB interface. When using RS-232-C, the HP Touchscreen (150) requires the use of separate files for output and input, only one of which may be open at a given time. Refer to Chapter 6 of the Operation and Interconnection Manual for an example. ■

ManualsPlus.com

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 allocates 5110 bytes of available graphics memory into the polygon buffer (sets other buffers to default allocations)

30 causes the program to pause until all of the bytes are allocated in the buffer by requesting that the plotter output the buffer size when it is empty

40 reads the output response generated by line 30 (the actual response is not important, but it should be read by the computer anyway; refer to Chapter 14 for hints on using the ESC.L instruction with the ESC.T instruction and for methods of reading the output response.)

50 initializes the plotter; selects pen 1; sets the initial pen position

60 clears the polygon buffer and enters polygon mode; defines three subpolygons consisting of progressively larger concentric circles

70 defines two more subpolygons consisting of larger concentric circles; exits polygon mode

80 fills alternate subpolygons using default fill conditions (solid, bi-directional with default PT.3 established by IN in line 50); edges the subpolygons

90 returns the pen to the carousel

The Polygon Buffer

When a polygon is created with the PM, RA, RR, EA, ER, WG, or EW instructions, the points defining the polygon are stored in the polygon buffer. When you want to fill or edge a polygon, the points are accessed from this buffer by the FP and EP instructions.

The size of the polygon buffer is determined by the first parameter in the GM or the second parameter in the ESC.T instruction. In many cases, you can leave the buffer at the default size of 1778 bytes. However, if you are defining a large polygon, you must first increase the buffer to avoid error 7, buffer overflow. Also, if you are not defining polygons and need extra bytes for one of the other buffers (e.g., the replot buffer), you might want to decrease the polygon buffer size. For information about the GM instruction, refer to The Graphics Memory Instruction, GM, in Chapter 3. For specifics on the ESC.T instruction, refer to The Allocate Configurable Memory Instruction, ESC.T, in Chapter 14.

ManualsPlus.com

Determining the Approximate Size of the Polygon Buffer

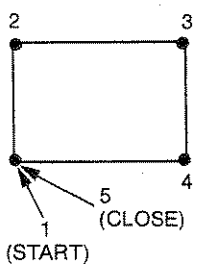
The size of the polygon buffer is specified in bytes. To determine how much space to set in the buffer, you must convert the number of points in your polygon into bytes. The following formula will suit most situations; however, it is approximate. If you need a more accurate determination, read the paragraphs titled Determining the Exact Size of the Polygon Buffer, later in this section.

$$\#\text{ of bytes} = \#\text{ of points} \times 14$$

The default size of the polygon buffer is 1778 bytes. This will accommodate approximately 127 points (1778/14).

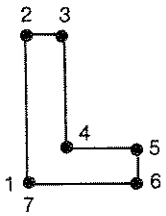
Counting the Points in Your Polygon

The starting pen position and each vertex defined with a coordinate pair are all points defining a polygon. Thus, although a rectangle has 4 corners, it consists of 5 points, as shown below.




```
"PU0,0;PM0;PD0,600,800,600,800,0,0,0;"

OR

"PM2;EP;"
```

The following shape has 7 points.




```
"PU0,0;PM0;PD0,800,200,800,200,200;"
"PD600,200,600,0,0,0;PM2;EP;"
```

The number of points in an arc or circle depends on the number of chords in the arc. You can use this formula:

$$\#\text{ of points} = \frac{\text{arc size (degrees)}}{\text{chord angle (degrees)}} + 1$$

Thus, a full circle with the default chord angle of 5 degrees consists of 73 points (360/5 + 1 = 73). On the other hand, a 45-degree arc with an angle of 3 degrees consists of 16 points (45/3 + 1 = 16).

ManualsPlus.com

Determining the Exact Size of the Polygon Buffer

To determine the exact number of bytes required for any polygon, use the equation shown next. Each segment of the equation is described in the following paragraphs; an example is provided at the end of this section.

$$\text{total \# of bytes} = 2 + [(p_1 + f_1) + (p_2 + f_2) + \ldots]$$

where

2 is the number of bytes required for overhead,

p is 1 byte caused by a PU, PD, PM 1, or PM 2 instruction, and

f is the following formula:

$$f = (12 \times \text{\# of points}) + \left(\left\lceil \frac{\text{\# of points}}{128} \right\rceil \times 2\right)$$

The ceiling symbol $\lceil\ \rceil$ indicates that the result of the division is to be rounded up to the next integer. Thus, for 1 point, $f = (12 \times 1) + (\lceil 1/128 \rceil \times 2) = 12 + 2 = 14$ (remember, $\lceil 1/128 \rceil = 1$, **not** 0.078). Similarly, for 4 points, $f = (12 \times 4) + (\lceil 4/128 \rceil \times 2) = 48 + 2 = 50$.

The easiest way to understand this equation is to try an example. Consider the polygon created for the surface chart in the example given previously under The Fill Polygon Instruction, FP:

```
PA 0,0;
PM 0;PD 0,10,10,16;
PD 20,20,30,14,40,18,50,16;
PD 60,22,60,0,0,0;PM 1;
PU 4,4;PD 4,8,16,8,16,4,4,4;PU;PM 2;
```

The total number of bytes will be determined as shown on the next page.

ManualsPlus.com

Overhead:	2
Current pen position before PM 0 (0 , 0) counts as 1 point, so solve *f* for 1 point:	14
PD causes 1 byte:	1
2 points follow PD, so solve *f* for 2 points:	26
PD causes 1 byte:	1
4 points follow PD, so solve *f* for 4 points:	50
PD causes 1 byte:	1
3 points follow PD, so solve *f* for 3 points:	38
PM 1* causes 1 byte:	1
PU causes 1 byte:	1
1 point follows PU, so solve *f* for 1 point:	14
PD causes 1 byte:	1
4 points follow PD, so solve *f* for 4 points:	50
PU causes 1 byte:	1
PM 2* causes 1 byte:	1
Total number of bytes required for this polygon:	202

Contrast this number with the previous general formula of 14 × # of points. This polygon has 15 points, so according to that formula, the polygon would require 210 bytes. By using the more precise equation, you save 8 bytes, which can be allocated to one of the other buffers in the configurable graphics memory. Of course, if you sent longer strings, thus eliminating the extra PD instructions, you would save a few more bytes. For example, sending all of the points between the first PD and the PM 1 in one string, you would save 6 more bytes.

*In this example, the last point before PM 1 closed the subpolygon, and was already included in the previous point count when solving for *f*. However, if you do not include a point that closes the subpolygon, the plotter will add this point, so you would need to add 14 (*f* for one point) after the PM 1. This also holds true for PM 2 if you do not include a point that closes the polygon.

ManualsPlus.com

ManualsPlus.com

Chapter 7
Labeling Basics

What You'll Learn in This Chapter

In this chapter you will learn how to label so that you can annotate plots or make text charts. You will learn how to position labels and how to change their size, slant, and direction. This chapter assumes that you are labeling only with the standard (default) character set, ANSI ASCII (English), and the standard font, fixed-space. However, all of the instructions in this chapter are valid for any character set and font. If you are interested in labeling with another character set, or with the variable-space font, read about the character set instructions in Chapter 11 after you have learned about labeling in this chapter.

HP-GL Instructions Covered

LB The Label Instruction
DT The Define Label Terminator Instruction
SI The Absolute Character Size Instruction
SR The Relative Character Size Instruction
SL The Character Slant Instruction
DI The Absolute Direction Instruction
DR The Relative Direction Instruction
LO The Label Origin Instruction
CP The Character Plot Instruction
ES The Extra Space Instruction
BL The Buffer Label Instruction
PB The Print Buffered Label Instruction
OL The Output Label Length Instruction

Terms You Should Understand

Fixed-Space Font — a set of characters where each character occupies a fixed, uniform amount of space. This is also called a vector font because the characters are drawn as a series of vectors (short line segments).

Variable-Space Font — a set of characters where each character occupies a variable (proportional) amount of space. This is also called an arc font

ManualsPlus.com

because the characters are drawn as a series of chords that approximate arc segments. This gives the characters a smoother appearance than vector-font characters.

Character Set — a group of characters, each of which is defined by a unique ASCII decimal code. Typically, a character set contains related characters, such as a character set composed of math symbols, or a Swedish character set. The HP 7550 has 20 character sets, each of which can be drawn in the fixed-space font, or the variable-space font.

Character Plot Cell — the concept used by the plotter to define a character, a line, and a space. The character plot cell is one space wide by one line high. The character occupies the lower-left portion of the cell, so that there is a blank area to the right and above the character.

Space — the width of the character plot cell; the space includes both the character and the blank area to the right. The default size of the space depends on the font; the average space is 1.5 times the average character width.

Line — the height of the character plot cell; the line includes both the character and the blank area above it. The default size of a line is 2 times the height of an uppercase A.

Character Origin — the lower-left corner of the character (i.e., the lower-left corner of the character plot cell).

Carriage-Return Point — the point to which the pen moves when it receives a carriage return (decimal code 13) in a label string.

Label Terminator — the final character in every label string. It terminates the label mode, so that the plotter will interpret subsequent characters as HP-GL instructions rather than as labeling characters. Its default value is the ASCII character **ETX** (decimal code 3), but it may be redefined using the DT instruction.

The Label Instruction, LB

USES: The LB instruction plots text using the currently defined character set. Use this instruction to annotate drawings or create text-only charts.

SYNTAX: *LB* c . . . c *term*
(where *term* is the label terminator defined by the DT instruction)

Parameter	Format	Range	Default
c . . . c	label	any character	none

ManualsPlus.com

EXPLANATION: All printing characters following the LB instruction are drawn using the currently defined character set. (Refer to Chapter 11 for defining character sets other than the default character set 0, ANSI ASCII English.) You can include nonprinting characters such as carriage return (**CR**, decimal code 13) and line feed (**LF**, decimal code 10); these characters are not drawn, but do cause the specified function to be performed.

NOTE: To terminate the LB instruction, you must use a special label terminator instead of the usual HP-GL terminator; if you don't use the label terminator, everything following the LB mnemonic will be printed, including other HP-GL instructions. The default label terminator is the nonprinting end-of-text character **ETX** (decimal code 3). You can change the label terminator with the define label terminator instruction, DT, described next in this chapter. ■

The label begins at the current pen position unless you have used the label origin instruction, LO, to specify the label's position. In addition, the label is plotted using the current direction, size, slant, and spacing. (Refer to the LO, DI, DR, SI, SR, SL, and ES instructions in this chapter for defaults and methods of changing these conditions.) After the label is drawn, the pen position is updated to be the next character origin (refer to Character Positioning later in this discussion).

It is possible to use the print buffered label instruction, PB, to repeat labels that have been drawn with the LB instruction. This is because the LB instruction clears the label buffer and places the first 150 characters in this buffer, at the same time drawing the label on your plot. The example later in this section shows how to use PB to repeat labels. However, the setting of the label origin (LO instruction) can limit the use of PB. If you are using label origins 1 through 3 or 11 through 13, you can repeat the last label any number of times. If you are using other label origins *and* the label contains an embedded carriage return (decimal code 13), only the part of the label following the carriage return will be repeated. In these situations, use the BL instruction instead of the LB instruction prior to printing the label with PB.

The label buffer is cleared by setting the plotter to default or initialized conditions, by executing a CM instruction, or by executing an LB or BL instruction without parameters.

NOTE: The label buffer is independent of the buffers in the configurable graphics memory. You cannot change the size of this buffer. ■

ManualsPlus.com

The following table summarizes the possible error conditions or unexpected results that you might observe with the LB instruction.

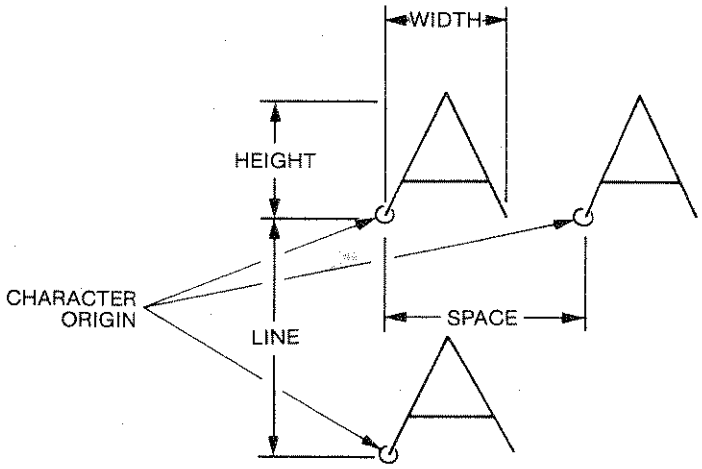
Condition	Error	Plotter Response
no parameter	none	clears label buffer
more than 150 characters	none	labels all characters, but places only the first 150 characters in the label buffer
label terminator invalid or omitted	none	labels all characters (including HP-GL mnemonics and parameters) until valid label terminator is encountered
lost mode	none	if location of label causes lost mode, conforms to conditions described under Relationship of Plotting Instructions and Graphics Limits in Chapter 4; if plotter is already in lost mode, ignores instruction
length of character string's baseline exceeds 2^{23} primitive grid units*	6	ignores instruction (plotter usually enters lost mode before this error occurs)

*Primitive grid units are the units the plotter uses to generate characters; refer to the User-Defined Character Instruction, UC, in Chapter 11. To clear the position-overflow condition, execute a UC or CP instruction without parameters, or any PA, PR, DI, DR, AA, AR, DF, or IN instruction. Pressing a cursor key on the front panel also clears the condition, as does a carriage return embedded in the label character string (assuming the plotter is not already in lost mode).

Character Positioning

The following illustration defines the relationships of a space, a line, the character origin, and the width and height of a character. The physical sizes of the space, line, width, and height depend on the current settings of the SI, SR, and ES instructions. The space size is the same for each character in the fixed-space fonts; in variable-space fonts, the space size varies with the width of each character. (For more details, refer to Adjusting Character Size, Spacing, and Position later in this chapter.)

ManualsPlus.com



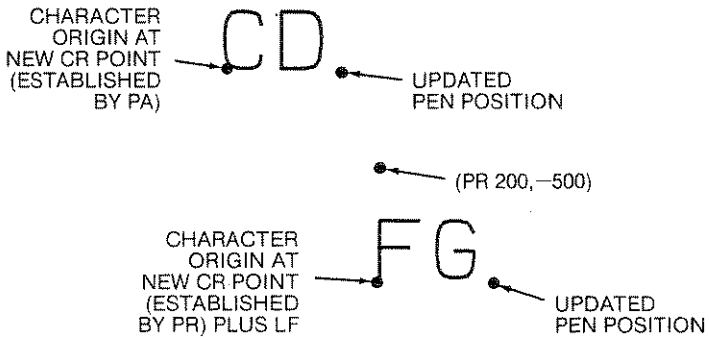
When you issue an LB instruction, the current pen position becomes the first character origin (unless altered by an LO instruction). After the first character is drawn, the pen moves to the next character origin and draws the next character (if any). This continues until the end of the label string, unless an embedded control character such as a carriage return (**CR**, CHR$(13)) or line feed (**LF**, CHR$(10)) is encountered. If a line feed is encountered, the pen moves down one line from its current position. If a carriage return is encountered, the pen moves to the carriage-return point (usually the current pen position when the LB instruction was executed, adjusted up or down by any line feeds or inverse line feeds). At the end of the string, the pen position is updated to be the character origin for the next character. These concepts are shown in the illustration on the next page.

NOTE: The carriage-return point is updated to be the current pen position after these instructions are executed: PA, PR, DI, DR, AA, AR, RO, DF, and IN. In addition, moving to a new point using the front-panel **P1**, **P2**, **ALIGN**, and cursor keys updates the carriage-return point to the new pen position. Also, the front-panel **CLEAR** and **RESET** function keys update the carriage-return point to the current pen position. Note that although the current pen position is updated after an LB instruction, the carriage-return point is not updated, except as described above. ■

ManualsPlus.com

```
"IN;SP1;SI.5,.8;"
"PA3010,4000;"
"LBAB"+CHR$(3)
```

```
"PA3010,2000;"
"LBCD"+CHR$(3)
"PR200,-500;"
"LB"+CHR$(13)+CHR$(10)+"FG"+CHR$(3)
```



How to Send the Label Terminator and Other Nonprinting Characters

In the HP-GL strings shown in the previous illustration, the characters **LF**, **CR**, and **ETX** all represent nonprinting characters. To determine whether you can produce these characters on your computer, check the ASCII character and key code table in your computer language documentation. On most HP computers, you can produce the **LF** character by pressing the **CTRL** (control) key followed by **J**. Likewise, the **CR** character is produced by **CTRL M**, and the **ETX** character is produced by **CTRL C**. If you cannot produce these characters from the keyboard, you will have to use a string function such as CHR$(*value*), which produces the character whose decimal code value is specified. For example, CHR$(3) produces **ETX**, CHR$(10) is **LF**, and CHR$(13) is **CR**. Again, check your computer language documentation for the proper function to use. (You can also refer to the programs for individual computers in Chapter 6 of the Operation and Interconnection Manual; they use string functions for labeling).

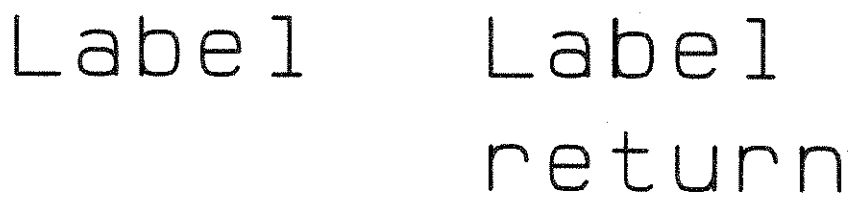
ManualsPlus.com

If you can produce the character directly from the keyboard, you can simply enter the character within the label instruction. However, if you must use a string function such as CHR$, you will have to send it separately from the string, as shown below. If necessary, substitute your computer's string function for CHR$. In addition, notice that the concatenation (linking) symbol + is used between the label string and the string function. This ensures that the character produced by the string function is sent immediately after the label string without any extra spaces. On some computers, & is used as the concatenation symbol; if this is true for your computer, substitute & for +.

Instruction Using String Function

```
PRINT #1, "IN;SP1;PA2000,5000;SI.55,.8;"
PRINT #1, "LBLabel"+CHR$(3)
PRINT #1, "PA4500,5000;"
PRINT #1, "LBLabel"+CHR$(13)+CHR$(10)+"return"+
          CHR$(3)
```

Whichever method you use to send the label and nonprinting characters, here is the plotted output:



Example — Repeating Labels from the Label Buffer

The following example illustrates the LB instruction, also showing that the contents of the label buffer can be drawn at different locations with the PB instruction. The labels are drawn using default conditions for the character set, size, direction, spacing, and slant. The PB instruction is described later in this chapter.

ManualsPlus.com

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP4;PA10,6000;"
30  PRINT #1, "LBLabel strings are buffered."+CHR$
              (13)+CHR$(10)+CHR$(3)
40  PRINT #1, "LBLast string is repeatable."+
              CHR$(3)
50  PRINT #1, "PA1000,5000;PB;"
60  PRINT #1, "PA500,4000;PB;"
70  PRINT #1, "SP0;"
80  END
```

```
Label strings are buffered.
Last string is repeatable.
```

```
Last string is repeatable.
```

```
Last string is repeatable.
```

The Define Label Terminator Instruction, DT

USES: The DT instruction specifies the character to be used as the label terminator. Use this instruction to define a new label terminator if the default label terminator (**ETX**, decimal code 3) cannot be used by your computer.

SYNTAX: *DT* label terminator *term*
or
DT *term*

Parameter	Format	Range	Default
label terminator	label	any character except **NULL**, **ENQ**, **LF**, **ESC**, and ; (decimal codes 0, 5, 10, 27, and 59, respectively)	**ETX** (decimal code 3)

EXPLANATION: The character following the DT mnemonic is interpreted to be the new label terminator. Label instructions react differently depending on which character you specify to be the label terminator.

ManualsPlus.com

LF or ; (decimal code 10 or 59)	Restores **ETX** (decimal code 3) as the label terminator.
printing character	Terminates the instruction and prints the character.
nonprinting character (control character)	Terminates the instruction and performs the function specified by the character.

The label terminator specified stays in effect until a new label terminator is specified, or the plotter is initialized or set to default conditions. All labeling instructions (LB, BL, and WD) following the DT instruction must be terminated by the specified character.

The following table summarizes the possible error conditions or unexpected results that you might observe with the DT instruction.

Condition	Error	Plotter Response
NULL or **ESC** characters specified	none	ignores instruction
LF or ; specified	none	restores default (**ETX**)
more than 1 character specified	2	executes first character

Example — Changing Label Terminators

The following program shows how to change the label terminator, as well as what happens to plotted labels with different terminators.

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP2;SC0,5000,0,5000;PA0,4500;"
30  PRINT #1, "LBDefault control character ETX"+CHR
      $(13)+CHR$(10)+CHR$(3)
40  PRINT #1, "LBterminates by performing end-"+CHR
      $(13)+CHR$(10)+CHR$(3)
50  PRINT #1, "LBof-text function."+CHR$(3)
60  PRINT #1, "PA0,3900;DT#;"
70  PRINT #1, "LBPrinting characters terminate,"+
      CHR$(13)+CHR$(10)+"#"
80  PRINT #1, "LBbut are also printed.#"
90  PRINT #1, "PA0,3400;DT"+CHR$(13)+";"
100 PRINT #1, "LBControl characters terminate"+CHR$
      (10)+CHR$(13)
110 PRINT #1, "LBand perform their function."+CHR$
      (13)+"SP0;"
120 END
```

ManualsPlus.com

```
Default control character ETX
terminates by performing end-
of-text function.

Printing characters terminate,
#but are also printed.#

Control characters terminate
and perform their function.
```

Labeling with Variables

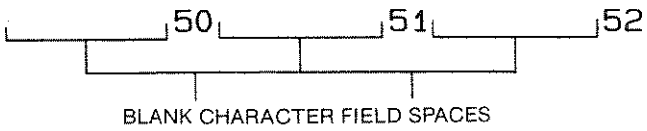
In some applications, it is desirable to label the plot using variables rather than literals to define the label string. The principles of sending variables in label instructions are similar to those for sending variables in plotting instructions (refer to Chapter 4). However, the format for the character field is important for a different reason — labels will be positioned differently and plotted with extra spaces, depending on the format you use. Many different conventions are used in various computer languages and computers to define variable length and the character field format in which these variables will be printed. To avoid unexpected placement of the labels defined by variables, refer to your computer manual for a definition of the conventions used to define the output character field.

Quotation marks are used by many computers to define the literal characters that are to be sent, but variables are not included within quotation marks. The comma is used by some computers as a delimiter between variables to cause the label string to be right-justified in a specific character-field width. The unused character positions in this field are normally sent as leading blank spaces to establish fixed spacing between label strings. For close spacing of label strings, the blank spaces can normally be suppressed by substituting a semicolon as a delimiter between variables.

The following example illustrates use of the comma to establish fixed spacing when using variables for labeling. When the value of X is 50, the labels shown are produced by the given HP-GL instructions. The second statement causes the plotter to label the values of X, X + 1, and X + 2. Blank spaces between the printed integers normally include space for the sign, which might not be printed, depending on your computer. The number of blank character-field spaces might vary with different computers.

ManualsPlus.com

```
X=50
"PA300,1300;LB",X,X+1,X+2,+CHR$(3)
```



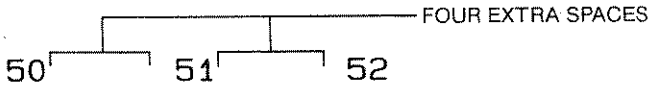
The following example illustrates the closer spacing achieved in Microsoft® BASIC when semicolons separate variables in labeling instructions. The semicolons between the variables cause suppression of blank spaces. The space between the printed integers varies with different computers, but normally includes the sign space.

```
X=50
"PA300,1000;LB";X;X+1;X+2;+CHR$(3)
```

```
50  51  52
```

Any spaces required to fit into the context of the item being labeled must normally be sent enclosed in quotation marks. The following example labels the same variables as above, but with four extra spaces between each of the integers. Note that four spaces enclosed in quotation marks are sent between each variable, but the semicolon suppresses unwanted blank spaces.

```
X=50
"PA300,700;LB";X;"    ";X+1;"    ";X+2;+CHR$(3)
```



Adjusting Character Size, Spacing, and Position

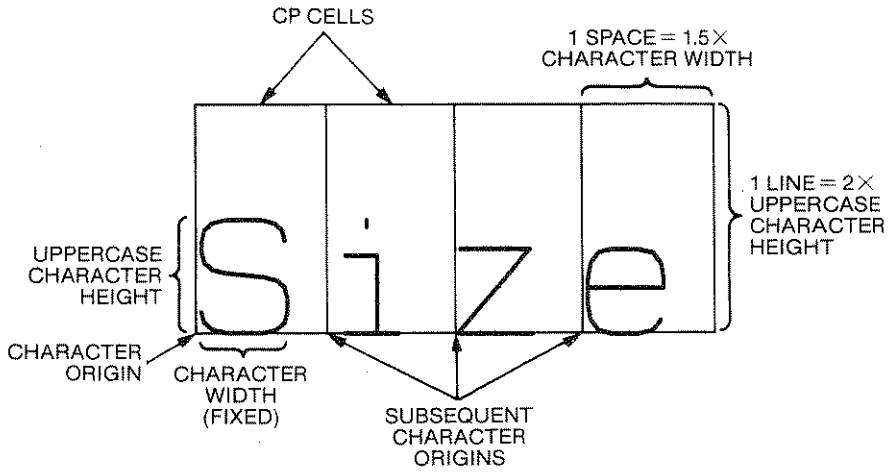
Now that you can plot labels, you are probably interested in learning how to adjust conditions such as size, slant, direction, space, and position. Most of these conditions are based on the concept of the "character plot cell," which is a rectangular grid used by the plotter to generate characters. The character plot cell is described in this section, followed by a summary of how various character conditions relate to this concept. The instructions for changing character conditions are presented in the remainder of this chapter. Note that these conditions affect both fixed-space and variable-space fonts. Refer to Chapter 11 for details on selecting fonts and character sets.

ManualsPlus.com

The Character Plot Cell

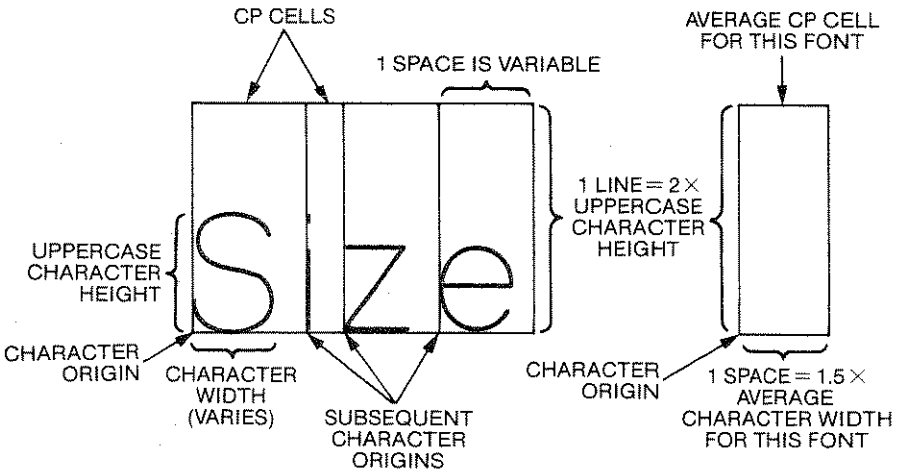
You can think of the character plot (CP) cell as a rectangular area around a character that includes blank areas above and to the right of the character, as shown below. Mapped onto this CP cell is a special coordinate system known as the "primitive grid;" the size of the grid (i.e., the number of grid units across the width and height) depends on the font selected. The grid isn't shown in the illustration, because it is only conceptual and changes with fonts. You will become better acquainted with the grid if you decide to design characters with the UC and DL instructions in Chapter 11.

The origin point of the grid (0,0) is also the character origin, and is always in the lower-left corner of the CP cell. The height of the character depends on the character size selected. However, the height of the CP cell (one line) is always 2 times the height of an uppercase A in the selected font. The width of the cell (one space) depends on whether a fixed-space font or a variable-space font is selected. If the font is fixed-space, the width of each character in the selected font is the same, and the CP cell is 1.5 times the width of a character. If the font is variable-space, the width of each character varies, as well as the width of the CP cell. Therefore, variable-space fonts have an "average" CP cell. The width of the average CP cell is 1.5 times the average character width.



Fixed-Space Font

ManualsPlus.com



Variable-Space Font

Character Size

The absolute character size and relative character size instructions, SI and SR, define the width and height of the characters in the font selected. If the font is fixed-space, the new width and height are applied to every character in the font. If the font is variable-space, the height is applied to every character, but the width is interpreted as an average width, so that each individual character width still varies, based on the average. The character size determines the size of the CP cell, as described earlier. Thus, whenever you change character size, you automatically change the CP cell size.

Direction and Slant

The absolute direction and relative direction instructions, DI and DR, define the orientation of the label on the page. For example, you can label horizontally (parallel to the X-axis), or you can label at any other angle. You can also change the slant of the characters with the character slant instruction, SL, to create an italicized effect. The direction and slant do not affect the size of the CP cell.

Spacing and Position

Default spacing is as described previously under The Character Plot Cell. To change spacing, use the extra space instruction, ES. This instruction does not affect the character size; it only allows you to add or subtract from the space around the character in the CP cell. You can adjust the vertical space (i.e., the line) and/or the horizontal space.

You can control the character position in a number of ways. Labels are drawn at the current pen position, so you can arrive at that position

ManualsPlus.com

with a plotting instruction such as PA, PR, AA, or AR. You can also define where the label is placed relative to the current pen position using the label origin instruction, LO (for example, centered, left-justified, or right-justified). Finally, you can use the character plot instruction, CP, to move the pen position in terms of the CP cell. This is useful for aligning labels. For example, when indenting to the third character of a previous label in a fixed-space font, you can use the CP instruction to move two CP cells, rather than having to measure that distance in current units.

It is sometimes tricky to position variable-space fonts, because the total length of the label depends on the widths of each character in the label. Usually you can use the LO instruction to position labels; the LO instruction automatically calculates the proper starting position for centered, left-, or right-justified labels, relative to the current pen position. Or, you can use the buffer label and output label length instructions, BL and OL, to determine the length of the label. (The length is also referred to as the baseline of the label.) Then you can use the length to calculate the position for the label.

The Absolute Character Size Instruction, SI

USES: The SI instruction specifies the size of characters in centimetres. Use this instruction to establish absolute character sizing that is not dependent on the settings of P1 and P2.

SYNTAX: *SI* width, height *term*
or
SI *term*

Parameter	Format	Range	Default
width	decimal	-2^{23} to $2^{23}-1$ centimetres*	0.285 cm (A3/B-size paper) 0.187 cm (A4/A-size paper)
height	decimal	-2^{23} to $2^{23}-1$ centimetres*	0.375 cm (A3/B-size paper) 0.269 cm (A4/A-size paper)

*excluding zero (0) and values approaching zero

EXPLANATION: An SI instruction without parameters (SI;) sets the character size to default values for the current paper size. The character size specified by SI is absolute, and is not affected by the changes in P1

ManualsPlus.com

or P2. The width is the actual character width for fixed-space fonts. For variable-space fonts, the width is the average character width. For all fonts, the height is the actual uppercase character height.

An SI instruction remains in effect until another SI or SR instruction is executed, or the plotter is initialized or set to default conditions.

The following table summarizes the possible error conditions or unexpected results that you might observe with the SI instruction.

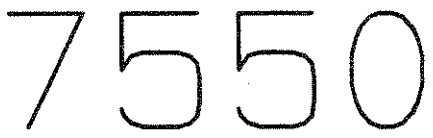
Condition	Error	Plotter Response
no parameters	none	establishes default values
1 parameter	2	ignores instruction
more than 2 parameters	2	executes first 2 parameters
either parameter is zero (0) or out-of-range	3	ignores instruction

Example — Changing Absolute Character Size

The following instructions draw the plotter's model number, 7550, in two different sizes: first, in the default size, and then 1-cm wide and 1.5-cm high.

```
"IN;SP1;PA100,800;"
"SI;LB7550"+CHR$(3)
"PA100,0;"
"SI1,1.5;LB7550"+CHR$(3)
```

7550



Example — Using Negative Parameters to Mirror Labels

Negative parameters produce mirror images of labels. A negative width parameter mirrors labels in the right-to-left direction.

```
"SI-.35,.6;LBHP"+CHR$(3)
```

A negative height parameter mirrors labels in the top-to-bottom direction.

```
"SI.35,-.6;LBHP"+CHR$(3)
```

ManualsPlus.com

Two negative parameters mirror labels in both directions, causing the label to appear to be rotated 180 degrees.

```
"SI-.35,-.6;LBHP"+CHR$(3)
```

dH

NOTE: The interactions of the SI, SR, DI, and DR instructions are complex when using negative parameters to produce mirror images. Refer to Parameter Interaction in Labeling Instructions at the end of this chapter for more examples of mirror images. ■

The Relative Character Size Instruction, SR

USES: The SR instruction specifies the relative size of characters as a percentage of the distance between scaling points P1 and P2. This means that character sizes are adjusted proportionally when P1 or P2 change positions. At power-on, the plotter assumes character sizes are relative. Use this instruction to change the percentage values for relative character sizes, or to reestablish relative character sizes after an absolute character size instruction, SI, has been executed.

SYNTAX: *SR* width, height *term*
or
SR *term*

Parameter	Format	Range	Default
width	decimal	-2^{23} to $2^{23}-1$ percentage*	0.75% of $\lvert P2_X - P1_X \rvert$
height	decimal	-2^{23} to $2^{23}-1$ percentage*	1.5% of $\lvert P2_Y - P1_Y \rvert$

*excluding zero (0) and values approaching zero

EXPLANATION: An SR instruction without parameters (SR;) sets the character size to default values. If P1 and P2 are at their default locations, this produces the same size characters (in centimetres) as listed previously under the SI instruction. However, with the SR instruction, the character size adjusts (expands or contracts) with subsequent changes in the locations of P1 and P2.

The character size specified by SR is a percentage of the distance in plotter units between the X- and Y-coordinates of P1 and P2. The plotter calculates the actual character width and height from the specified width and height parameters as follows:

$$\text{actual character width} = (\text{width}/100) \times \lvert P2_X - P1_X \rvert$$

$$\text{actual character height} = (\text{height}/100) \times \lvert P2_Y - P1_Y \rvert$$

ManualsPlus.com

Thus, if you have default P1 and P2 settings for A4/A-size paper and specify a width of 2% and a height of 3.5%, the actual character width would be (2/100) × (10 000) = 200 plotter units (or 0.5 cm). The actual character height would be (3.5/100) × (7200) = 252 plotter units (or 0.63 cm). If you changed the P1 setting to 100,100 and the P2 setting to 5000,5000 but didn't change the SR parameters, the new character width would be (2/100) × (5000 − 100) = 98 plotter units (or 0.245 cm). The new character height would be (3.5/100) × (5000 − 100) = 171.5 plotter units (or 0.429 cm).

As with the SI instruction, the width is the actual character width for fixed-space fonts. For variable-space fonts, the width is the average character width. For all fonts, the height is the actual uppercase character height.

NOTE: Either negative SR parameters or switching the relative positions of P1 and P2 will produce mirror images of labels. Refer to The Absolute Size Instruction, SI, and Parameter Interaction in Labeling Instructions in this chapter for more information on mirror images. ■

An SR instruction remains in effect until another SR or SI instruction is executed, or the plotter is initialized or set to default conditions.

The following table summarizes the possible error conditions or unexpected results that you might observe with the SR instruction.

Condition	Error	Plotter Response
no parameters	none	establishes default values
1 parameter	2	ignores instruction
more than 2 parameters	2	executes first 2 parameters
either parameter is zero (0) or out-of-range	3	ignores instruction

Example — Changing Relative Character Size

The instructions on the next page show how changes in P1 and P2 affect labels drawn while an SR instruction is in effect. The first line initializes the plotter, which establishes default P1 and P2 settings along with default relative character size. Then the first label is drawn. Next P1 and P2 are changed to define a square area, and a new label is drawn with the default character size. Finally, the character size is changed without changing P1 and P2, and another label is drawn. Notice that the new character size has equal parameters of 3% each; because the P1/P2 area is square, the characters are square. The plotted result was drawn on A4/A-size paper.

ManualsPlus.com

```
"IN;SP2;PA100,7000;LBDEFAULT SIZE"+CHR$(3)
"IP0,0,5500,5500;PA100,6500;"
"LBNEW P1 AND P2 CHANGE LABEL SIZE"+CHR$(3)
"SR3,3;PA100,6000;"
"LBNEW SR INSTRUCTION"+CHR$(13)+CHR$(10)+"CHANGES
  LABEL SIZE"+CHR$(3)
"SP0;"
```

DEFAULT SIZE

NEW P1 AND P2 CHANGE LABEL SIZE

NEW SR INSTRUCTION CHANGES LABEL SIZE

The Character Slant Instruction, SL

USES: The SL instruction specifies the slant at which characters are drawn. Use this instruction to create slanted text for emphasis.

SYNTAX: *SL* tan θ *term*
or
SL *term*

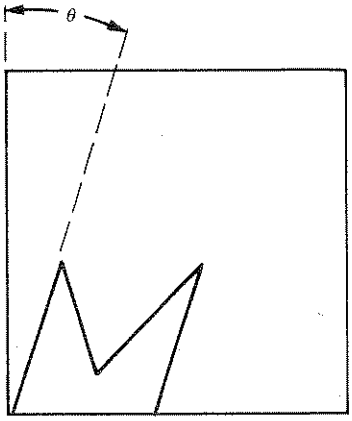
Parameter	Format	Range	Default
tangent θ	decimal	± 0.05 to ± 2 for default characters* ± 0.05 to ± 3.5 for large characters*	0 (no slant)

*These are practical ranges; the plotter actually accepts the full range of -2^{23} to $2^{23}-1$.

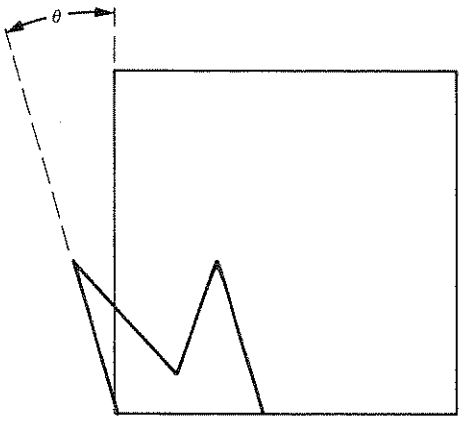
EXPLANATION: An SL instruction without parameters (SL;) sets the character slant to the default value of no slant. This is the same as an SL instruction with a parameter of zero (SL0;). The settings of P1 and P2 do not have any effect on the slant.

The parameter is the tangent of the angle θ from vertical, as shown on the next page. The base of the character always stays on the horizontal.

ManualsPlus.com



Positive Slant



Negative Slant

You can specify the actual tangent value (in degrees), or you can use the TAN function available on most computers (send the TAN function to the plotter as you would send a variable). Both methods are shown in the example at the end of this section. A table of tangent values for selected angles from −90 to 90 degrees follows. Note that the tangent of ±90 is infinity, and will cause a computer error on most systems.

θ	Tangent	θ	Tangent
0	0	0	0
−10	−0.18	10	0.18
−20	−0.36	20	0.36
−30	−0.58	30	0.58
−40	−0.84	40	0.84
−45	−1.00	45	1.00
−50	−1.19	50	1.19
−60	−1.73	60	1.73
−70	−2.75	70	2.75
−80	−5.67	80	5.67
−90	infinity	90	infinity

A practical parameter range for this instruction is ±.05 to ±1.5. An SL instruction remains in effect until another SL instruction is executed, or the plotter is initialized or set to default conditions.

ManualsPlus.com

The following table summarizes the possible error conditions or unexpected results that you might observe with the SL instruction.

Condition	Error	Plotter Response
no parameters	none	establishes no slant (vertical characters)
more than 1 parameter	2	executes first parameter
parameter out-of-range	3	ignores instruction

Example — Specifying Character Slant

The following example shows two methods of specifying slant. The first label is drawn at a slant of 20 degrees by using a variable generated by the TAN function. The second uses the tangent value given in the previous table to draw at a slant of −20 degrees.

NOTE: If you want to use the TAN function, check your documentation to see how your computer interprets angles. Microsoft® BASIC interprets angles as radians, so line 40 in this program converts degrees to radians. On the HP Series 200 computers, simply execute the BASIC statement DEG before using the TAN function. ■

```
10 'Insert configuration statement here
20 PRINT #1, "IN;SP1;SI1,1.5;PA500,3000;"
30 PI=3.141593
40 A=TAN(20*(PI/180))
50 PRINT #1, "SL";A;";LBSlant"+CHR$(3)
60 PRINT #1, "PA500,2000;SL-.36;LBSlant"+CHR$(3)
70 PRINT #1, "SP0;"
80 END
```

ManualsPlus.com

The Absolute Direction Instruction, DI

USES: The DI instruction specifies the direction in which labels are drawn, independent of P1 and P2 settings. Use this instruction to change labeling direction when you want to label a curve in a line chart or label schematic drawings and blueprints.

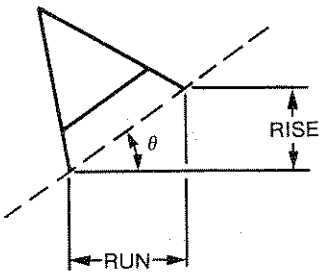
SYNTAX: *DI* run, rise *term*
or
DI *term*

Parameter	Format	Range	Default
run (or cos θ)	decimal	-2^{23} to $2^{23}-1$	1
rise (or sin θ)	decimal	-2^{23} to $2^{23}-1$	0

EXPLANATION: A DI instruction without parameters (DI;) sets the label direction to horizontal (parallel to the X-axis). The settings of P1 and P2 do not have any effect on the label direction.

You can express the parameters as the run and rise, or using trigonometric functions cos and sin according to the following relationship:

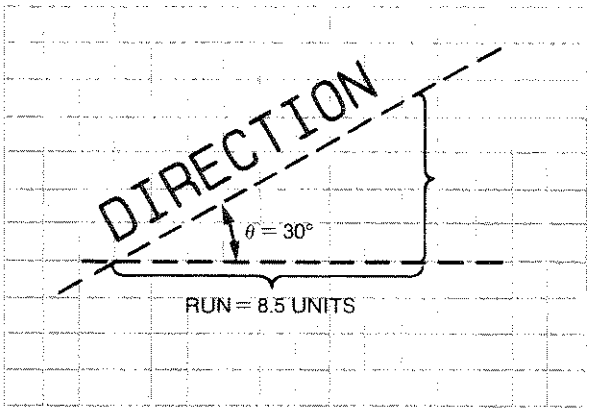
where: $\theta = \tan^{-1}\left(\frac{\text{rise}}{\text{run}}\right)$ and $\tan\theta = \frac{\sin\theta}{\cos\theta} = \frac{\text{rise}}{\text{run}}$



Suppose you want your label to be plotted in the direction shown in the following graph. You can do this in either of two ways: measure the run and the rise, or the angle. To use the first method, extend lines along the label and parallel to the X-axis. If you measure the run and rise, you can use these as the parameters of the DI instruction (DI 8.5 , 4.9 ;).

Or, if you know the angle, you can use the trigonometric values (since sin/cos = rise/run). In this example, $\theta = 30°$; cos 30° = 0.866, and sin 30° = 0.5. Thus, you can use these as parameters of the DI instruction (DI .866 , .5 ;). Whichever set of parameters you use, the label will be drawn in the same direction as shown in the following figure.

ManualsPlus.com



DI 8.5, 4.9;
or
DI .866, .5;

If you know the angle, you can specify the actual cosine and sine values, or you can use the SIN and COS functions available on most computers (send these functions to the plotter as you would send a variable). Both methods are shown in the example at the end of this section. A table of cosine values and sine values for selected angles from −360 to 360 degrees follows.

θ	Cosine	Sine
0	1	0
−30	0.87	−0.50
−45	0.71	−0.71
−60	0.50	−0.87
−90	0	−1
−120	−0.50	−0.87
−135	−0.71	−0.71
−150	−0.87	−0.50
−180	−1	0
−210	−0.87	0.50
−225	−0.71	0.71
−240	−0.50	0.87
−270	0	1
−300	0.50	0.87
−315	0.71	0.71
−330	0.87	0.50
−360	1	0

θ	Cosine	Sine
0	1	0
30	0.87	0.50
45	0.71	0.71
60	0.50	0.87
90	0	1
120	−0.50	0.87
135	−0.71	0.71
150	−0.87	0.50
180	−1	0
210	−0.87	−0.50
225	−0.71	−0.71
240	−0.50	−0.87
270	0	−1
300	0.50	−0.87
315	0.71	−0.71
330	0.87	−0.50
360	1	0

ManualsPlus.com

At least one parameter must be nonzero. For example:

DI Instruction	Label Direction
DI 1,0	horizontal
DI 0,1	vertical
DI 1,1 or DI 0.7,0.7 (any parameters equal to each other)	45-degree angle

A DI instruction remains in effect until another DI or DR instruction is executed, or the plotter is initialized or set to default conditions. A DI instruction updates the carriage-return point to the current pen position.

The following table summarizes the possible error conditions or unexpected results that you might observe with the DI instruction.

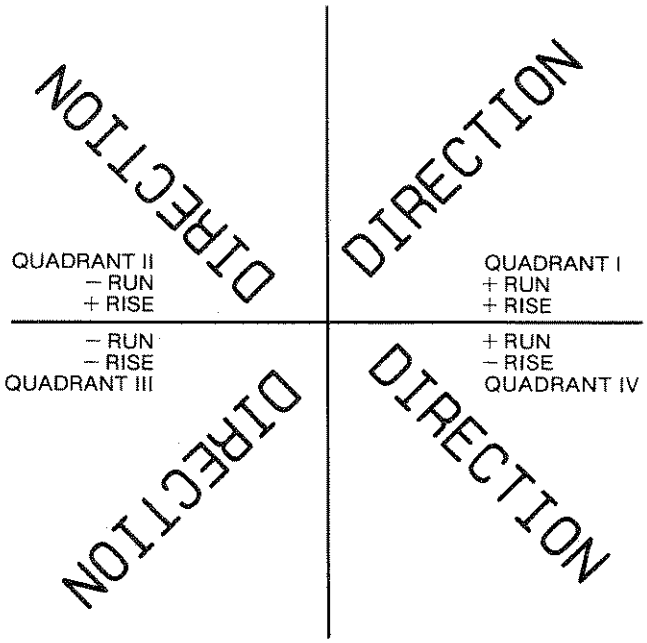
Condition	Error	Plotter Response
no parameters	none	establishes horizontal label direction
more than 2 parameters	2	executes first 2 parameters
1 parameter	2	ignores instruction
parameter out-of-range	3	ignores instruction
both parameters are zero (0)	3	ignores instruction

Examples — Rotating Label Direction

The size and sign of the two parameters in the DI instruction determine the amount of rotation. In the following example, a label is drawn four times using each possible combination of positive and negative parameters. If you imagine the current pen position to be the origin of a coordinate system, you can see that the signs of the parameters determine which quadrant the label will be in.

```
"IN;SP1;SI.3,.5;"
"PA3000,3000;DI1,1;LB  DIRECTION"+CHR$(13)+CHR$(3)
"DI1,-1;LB  DIRECTION"+CHR$(13)+CHR$(3)
"DI-1,-1;LB  DIRECTION"+CHR$(13)+CHR$(3)
"DI-1,1;LB  DIRECTION"+CHR$(13)+CHR$(3)
"SP0;"
```

ManualsPlus.com

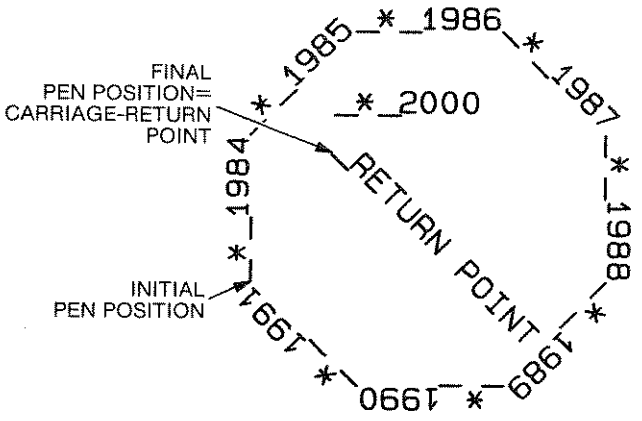


The next instructions label the years 1984 through 1991, in a circular pattern starting with a vertical label. The direction in which each year is labeled is changed in increments of 45 degrees. Then the labels in the center are drawn to illustrate the use of the COS and SIN functions as parameters. The label _*_2000 contains both a carriage return, CHR$(13), and a line feed, CHR$(10), before the label terminator, CHR$(3), so the pen position at the end of that label is one line below the beginning of that label. You can see that DI instructions update the carriage-return point by observing the pen's position at the end of the program. The final character in the last label is a carriage return, and the pen returns to the carriage-return point, the position of the pen at the last DI instruction.

NOTE: If you want to use the COS and SIN functions on your computer, check your documentation to see how your computer interprets angles. Microsoft® BASIC interprets angles as radians, so lines 130–140 and 170–180 convert degrees to radians when using the COS and SIN functions. On the HP Series 200 computers, simply execute the BASIC statement DEG when using the SIN and COS functions. ■

ManualsPlus.com

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA6000,3000;"
30  PRINT #1, "DI0,1;LB_*_1984"+CHR$(3)
40  PRINT #1, "DI1,1;LB_*_1985"+CHR$(3)
50  PRINT #1, "DI1,0;LB_*_1986"+CHR$(3)
60  PRINT #1, "DI1,-1;LB_*_1987"+CHR$(3)
70  PRINT #1, "DI0,-1;LB_*_1988"+CHR$(3)
80  PRINT #1, "DI-1,-1;LB_*_1989"+CHR$(3)
90  PRINT #1, "DI-1,0;LB_*_1990"+CHR$(3)
100 PRINT #1, "DI-1,1;LB_*_1991"+CHR$(3)
110 PRINT #1, "PA6450,3900;"
120 PI=3.141593
130 A=COS(0*(PI/180))
140 B=SIN(0*(PI/180))
150 PRINT #1, "DI";A;",";B;";"
160 PRINT #1, "LB_*_2000"+CHR$(13)+CHR$(10)+CHR$(3)
170 C=COS(-45*(PI/180))
180 D=SIN(-45*(PI/180))
190 PRINT #1, "DI";C;",";D;";"
200 PRINT #1, "LB_RETURN POINT"+CHR$(13)+CHR$(3)
210 PRINT #1, "SP0;"
220 END
```



The Relative Direction Instruction, DR

USES: The DR instruction specifies the direction in which labels are drawn, relative to the scaling points P1 and P2. *This means that label direction is adjusted when P1 and P2 change so that labels maintain the same relationship to the plotted data.* At power-on, the plotter assumes label direction is relative. Use this instruction to change the percentage values for relative label direction, or to reestablish relative direction after a DI instruction has been executed.

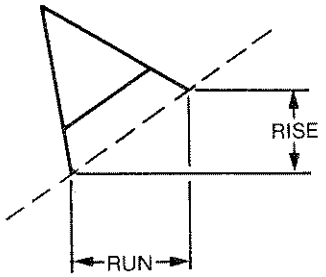
ManualsPlus.com

SYNTAX: *DR* run, rise *term*
or
DR *term*

Parameter	Format	Range	Default
run	decimal	-2^{23} to $2^{23}-1$	1% of $P2_X - P1_X$
rise	decimal	-2^{23} to $2^{23}-1$	0% of $P2_Y - P1_Y$

EXPLANATION: A DR instruction without parameters (DR;) sets the label direction to horizontal (parallel to the X-axis). When you include parameters, the actual angle at which the label is plotted will vary with the settings of P1 and P2. At least one parameter must be nonzero.

You can express the parameters as the run and rise according to the following relationship:



where: run — interpreted as a percentage of $P2_x - P1_x$

rise — interpreted as a percentage of $P2_y - P1_y$

The concept of specifying the run and the rise is similar to the DI instruction (refer to the discussion under The Absolute Direction Instruction, DI, described previously.) However, there is a very important distinction between the DR and DI instructions. In the DR instruction, the run and rise parameters are a *percentage* of the X- and Y-distances between P1 and P2, as shown below. As a result, when the settings of P1 and P2 change, the percentage remains constant and the *angle* changes. The examples at the end of this section show how relative label direction changes when P1 and P2 are changed.

A DR instruction remains in effect until another DR or DI instruction is executed, or the plotter is initialized or set to default conditions.

The following table summarizes the possible error conditions or unexpected results that you might observe with the DR instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	establishes horizontal label direction
more than 2 parameters	2	executes first 2 parameters
1 parameter	2	ignores instruction
parameter out-of-range	3	ignores instruction
both parameters are zero (0)	3	ignores instruction

Example — Effects of Changing P1 or P2 on Relative Label Direction

The following description and program will help you visualize the direction of labeling when using the DR instruction with various parameters.

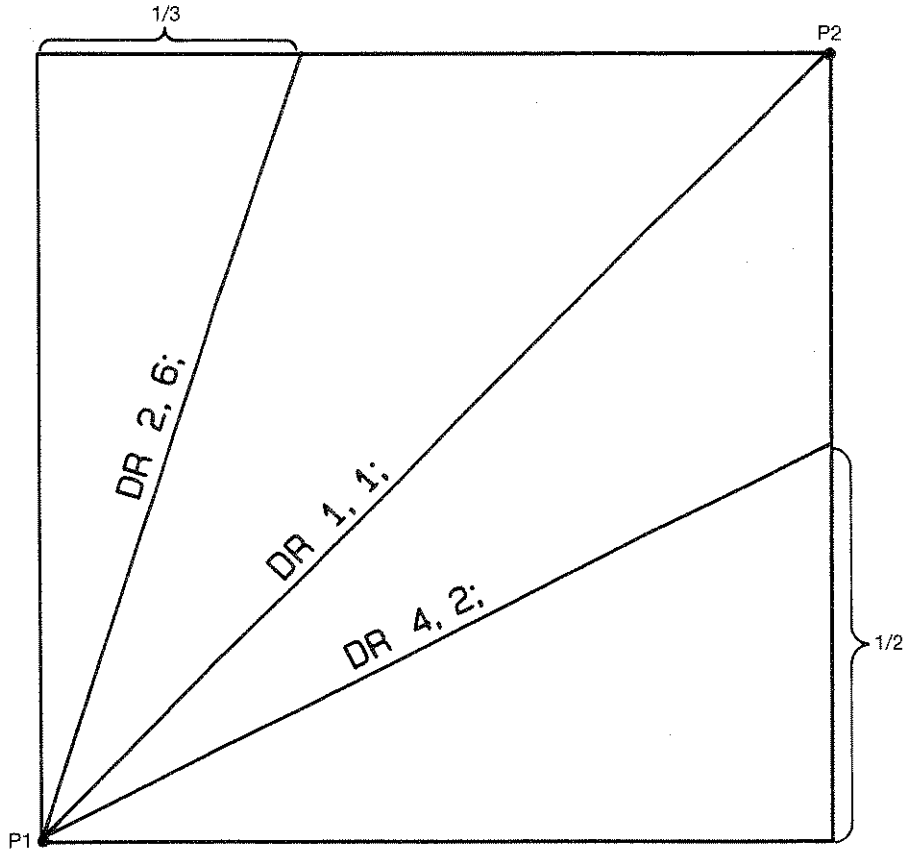
Think of label directions as being parallel to a line starting at the lower-left scaling point (usually P1) and intersecting the top edge or the opposite side of the P1/P2 area. To calculate where the intersection is, determine which is larger — the run or the rise.

- If run = rise, the directional line will go directly from P1 to P2.
- If rise > run, the directional line will intersect the top of the plotting area, a fraction of the way across toward P2. The fraction is determined by run/rise. If run = 2 and rise = 6, the line intersects the top of the plotting area one-third (2/6) of the way toward P2.
- If run > rise, the directional line will intersect the side of the plotting area, a fraction of the way up toward P2. The fraction is determined by rise/run. If run = 4 and rise = 2, the line intersects the side of the plotting area one-half (2/4) of the way toward P2.

Remember, since labeling starts at the current pen position, labels will be parallel to these directional lines, not necessarily on them. Also, negative parameters have the same effect on direction as described under the DI instruction.

The following example illustrates relative label directions.

ManualsPlus.com




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;IP0,0,4800,4800;"
30  PRINT #1, "SC0,1000,0,1000;SR2,3;"
40  PRINT #1, "PA1000,1000;EA0,0;"
50  PRINT #1, "PD0,0;PU;DR1,1;"
60  PRINT #1, "CP15,.25;LBDR 1,1;"+CHR$(3)
70  PRINT #1, "PU333.33,1000;PD0,0;PU;DR2,6;"
80  PRINT #1, "CP15,.25;LBDR 2,6;"+CHR$(3)
90  PRINT #1, "PU1000,500;PD0,0;PU;DR4,2;"
100 PRINT #1, "CP15,.25;LBDR 4,2;"+CHR$(3)
110 PRINT #1, "SP0;"
120 END
```

ManualsPlus.com

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; sets the scaling points P1 and P2 to define a square area

30 scales the P1/P2 area into user units; sets a relative character size

40 moves to P2; outlines the P1/P2 area

50 draws a line from P2 to P1 and lifts the pen; establishes a relative label direction

60 moves the pen 15 CP-cell spaces and 0.25 CP-cell lines (the CP instruction is described later in this chapter); labels "DR 1 , 1 ;"

70 draws a line between P1 and a point that is one-third of the way to P2 along the top of the P1/P2 area and lifts the pen; establishes a new relative label direction

80 moves the pen 15 CP-cell spaces and 0.25 CP-cell lines; labels "DR 2 , 6 ;"

90 draws a line between P1 and a point that is one-half of the way to P2 along the side of the P1/P2 area and lifts the pen; establishes a new relative label direction

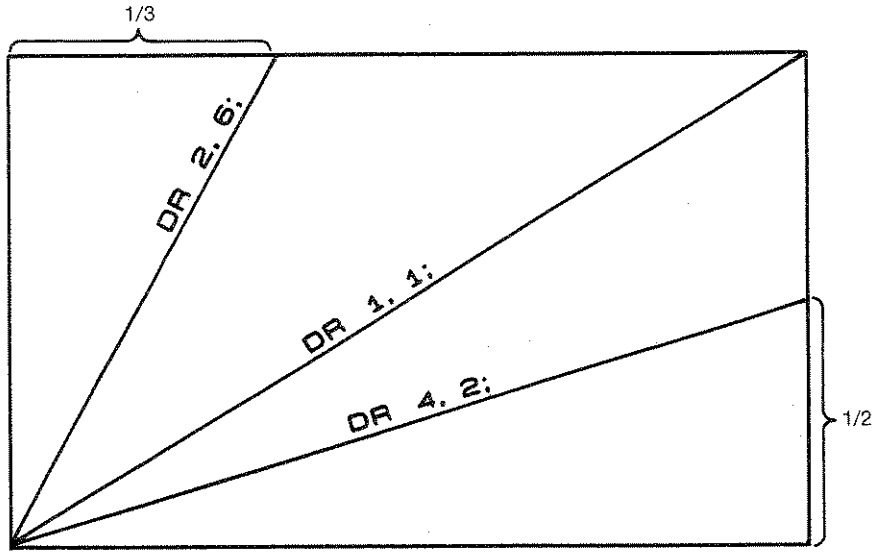
100 moves the pen 15 CP-cell spaces and 0.25 CP-cell lines; labels "DR 4 , 2 ;"

110 returns the pen to the carousel

Now change the relative positions of P1 and P2 by replacing line 20 in the previous program with the line shown on the next page. Notice that each directional line is drawn the same fraction of the way toward P2; however, the angles of the lines have changed in order to maintain the correct relative direction. Notice also that the character size has changed because a relative size was specified in line 30.

ManualsPlus.com

```
20  PRINT #1, "IN;SP1;IP5000,0,9800,3000;"
```



The Label Origin Instruction, LO

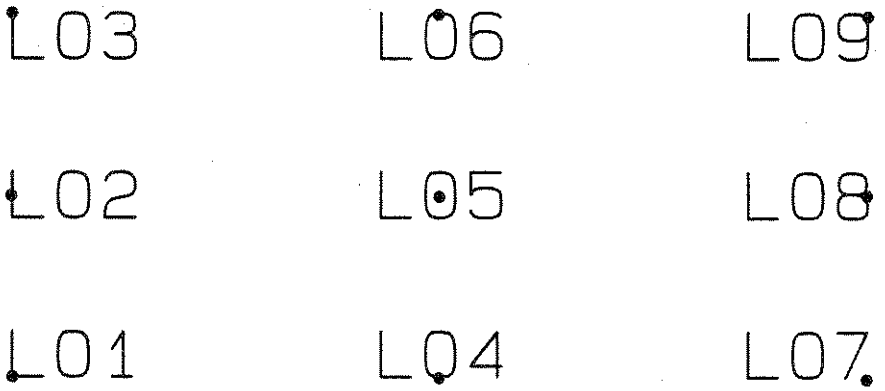
USES: The LO instruction positions labels relative to the current pen position. Use this instruction to center labels or justify them to the left or right of the current pen position. Positioning can be above or below the current pen position and can also be offset by an amount equal to 1/2 the character's width and height.

SYNTAX: *LO* position number *term*
or
LO *term*

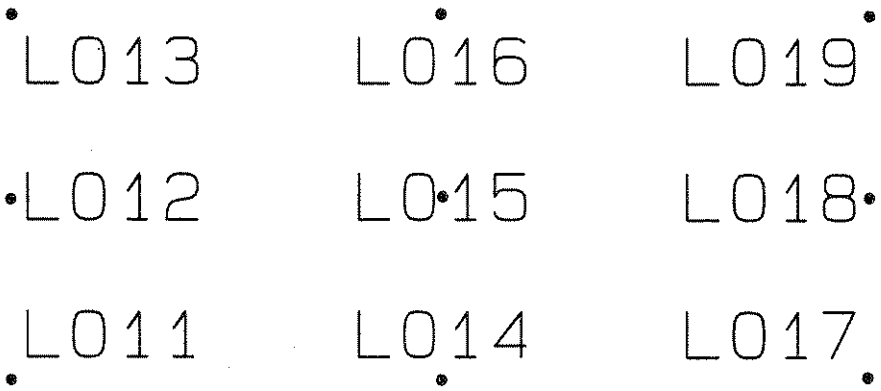
Parameter	Format	Range	Default
position number	integer	1-9 or 11-19	1

EXPLANATION: The position number determines the position of the label with respect to the current pen position. The illustration on the following page summarizes this relationship by showing the instructions LO 1 through LO 9 in the labeled position that they produce. Each dot represents the current pen position. An LO instruction without parameters (LO;) sets the default label origin, LO 1.

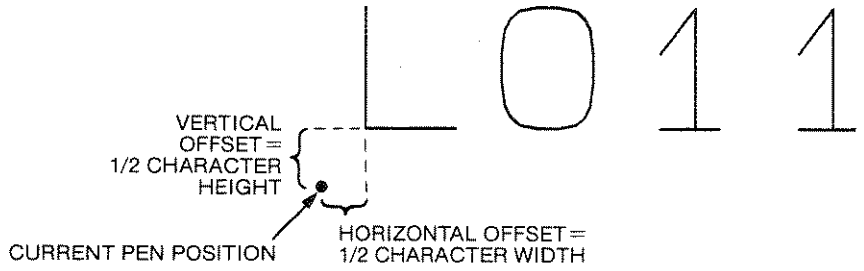
ManualsPlus.com



The positions resulting from the instructions LO 11 through LO 19 are the same as shown for the instructions LO 1 through LO 9, except that the labels are offset from the current pen position as shown below.



The amount of offset is equal to 1/2 of the character's width and 1/2 of the character's height. (The width and height are specified by the most recent SI or SR instruction.) The offset is shown below.



An LO instruction remains in effect until another LO instruction is executed, or the plotter is initialized or set to default conditions.

If position numbers 1 through 3 or 11 through 13 are specified, positioning calculations are performed on individual characters and the

ManualsPlus.com

characters are drawn as they are received and stored in the label buffer. However, if position numbers 4 through 9 or 14 through 19 are specified, positioning calculations are not performed until the length of the label string is established by receipt of either a carriage return or the label terminator. In these cases, the characters are stored in the label buffer and the label string is not drawn until a carriage return or label terminator is received. The capacity of the label buffer limits line length for these centered or right-justified labels to 150 characters, including control characters and the carriage return or label terminator.

The pen position is updated after each character is drawn, and the pen automatically moves to the next character origin in anticipation of additional characters. If you wish to return the pen to its location prior to the label instruction, you can send a carriage return following the label string and before the label terminator.

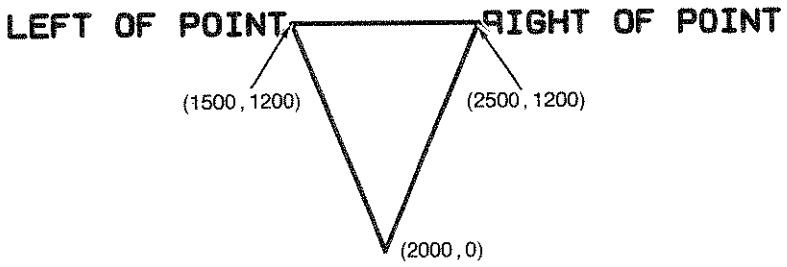
The following table summarizes the possible error conditions or unexpected results that you might observe with the LO instruction.

Condition	Error	Plotter Response
no parameters	none	establishes label origin 1
more than 1 parameter	2	executes first parameter
parameter out-of-range	3	ignores instruction

Examples — Positioning Labels with LO

The following example demonstrates the use of the carriage-return with an LO instruction.

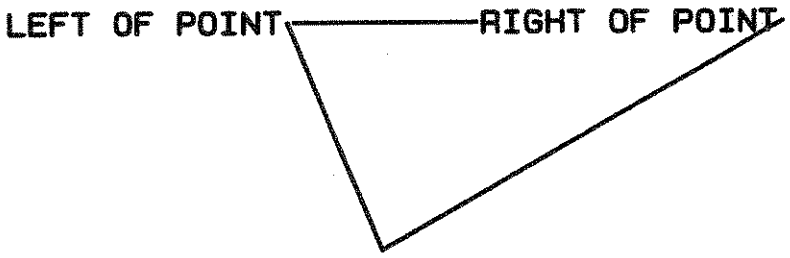
```
"IN;SP1;PA2000,0;"
"PD1500,1200;LO18;LBLEFT OF POINT"+CHR$(13)+CHR$(3)
"PD2500,1200;LO12;LBRIGHT OF POINT"+CHR$(13)+CHR$(3)
"PD2000,0;SP0;"
```



The plot on the following page shows the effect on the pen position when you do not include the carriage return in the previous two LB instructions.

ManualsPlus.com

```
"IN;SP1;PA2000,1500;"
"PD1500,2700;LO18;LBLEFT OF POINT"+CHR$(3)
"PD2500,2700;LO12;LBRIGHT OF POINT"+CHR$(3)
"PD2000,1500;SP0;"
```



When you divide a label by embedding carriage returns in an LB label string, each portion of the label is positioned according to the label origin. Refer to the following example.

```
"IN;SP1;PA2000,6000,LO14;"
"LBEMBEDDED"+CHR$(13)+CHR$(10)+"CARRIAGE-RETURN"+
   CHR$(13)+CHR$(10)+"CHARACTERS"+CHR$(3)
"SP0;"
```

EMBEDDED
CARRIAGE-RETURN
CHARACTERS

The Character Plot Instruction, CP

USES: The CP instruction moves the pen the specified number of character plot cells. Use this instruction to move the pen any number of character spaces or lines from the current pen position. For example, you can move the pen any number of character spaces in order to indent a label.

SYNTAX: *CP* spaces, lines *term*
or
CP *term*

Parameter	Format	Range	Default
spaces	decimal	-2^{23} to $2^{23}-1$	none
lines	decimal	-2^{23} to $2^{23}-1$	none

EXPLANATION: A CP instruction without parameters (CP;) performs a carriage return and line feed, moving one line down and returning to the carriage-return point. When specified, the parameters are

ManualsPlus.com

interpreted as follows. (For more information on spaces, lines, and the character plot cell, refer to Adjusting Character Size, Spacing, and Position earlier in this chapter.)

1. Spaces. A space is defined as the width of one character plot cell (or average character plot cell, for variable-space fonts), including any adjustments made by the extra space instruction, ES (described next in this chapter).

   The width of the character plot cell is 1.5 times the character width, which is specified in the SI or SR instructions. Thus, for "SI w,h;" and "ES s,l;", the formula for a CP space is $(1+s) \times (1.5 \times w)$ cm (where w = width, h = height, s = space, and l = line).

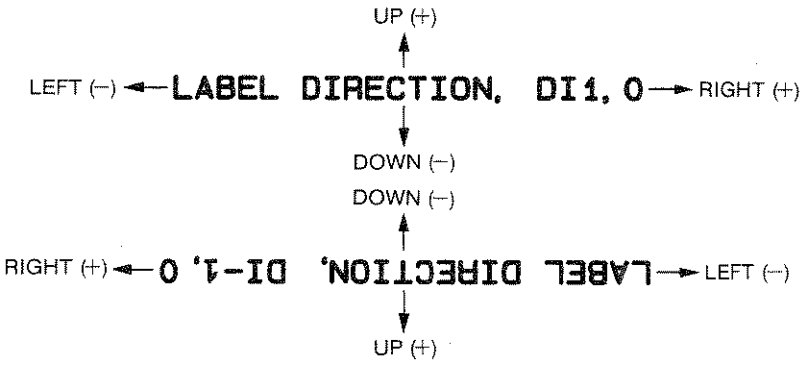
   Positive space values specify the number of spaces the pen will move to the right of the current pen position; negative values specify the number of spaces the pen will move to the left. Refer to the illustration following the description of the lines parameter.

2. Lines. A line is defined as the height of one character plot cell (or average character plot cell, for variable-space fonts), including any adjustments made by the extra space instruction, ES (described next in this chapter).

   The height of the character plot cell is 2 times the uppercase character height, which is specified in the SI or SR instructions. Thus, for "SI w,h;" and "ES s,l;", the formula for a CP line is $(1+l) \times (2 \times h)$ cm (where w = width, h = height, s = space, and l = line).

   Positive line values specify the number of lines the pen will move up from the current pen position; negative values specify the number of lines the pen will move down. A CP instruction that has a nonzero lines parameter shifts the carriage-return point up or down by the amount specified.

Note that the right, left, up, and down movements of positive and negative parameters are relative to the label direction, as shown here.



ManualsPlus.com

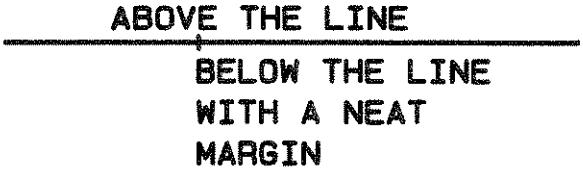
CP instructions are executed with the current pen status (up or down), and all moves are made with respect to the current character origin. If an LO instruction is in effect, a pen-up move is made to the label origin before the CP instruction is executed. The CP instruction affects only the placement of the next label; you must issue new CP instructions to affect subsequent labels.

The following table summarizes the possible error conditions or unexpected results that you might observe with the CP instruction.

Condition	Error	Plotter Response
no parameters	none	performs carriage return and line feed
1 parameter	2	ignores instruction
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction

Example — Using CP to Align Labels

This example uses the CP instruction to produce lettering along a line (but not directly on top of it) and to align labels along a left margin.




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA4000,7000;PD1000,7000;PU;"
30  PRINT #1, "CP5,.35;LBABOVE THE LINE"+CHR$(3)
40  PRINT #1, "PA2000,7000;XT;CP0,-.95;"
50  PRINT #1, "LBBELOW THE LINE"+CHR$(13)+CHR$(10)
              +"AND WITH A NEAT"+CHR$(3)
60  PRINT #1, "CP;LBMARGIN"+CHR$(3)
70  PRINT #1, "SP0;"
80  END
```

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; draws a line 3000 plotter units long, ending at the point 1000,7000 (this is the starting position and the carriage-return point for the label in line 30); lifts the pen

ManualsPlus.com

30 moves 5 CP-cell spaces to the right and 0.35 CP-cell lines up; labels "ABOVE THE LINE"

40 moves to the point 2000 , 7000 (this is the new starting position and carriage-return point for the label in line 40); draws an X-tick to mark the new carriage-return point; moves 0.95 CP-cell lines down

50 labels "BELOW THE LINE"; uses embedded carriage return and line feed to move to the carriage-return point and down 1 CP-cell line, and labels "AND WITH A NEAT"

60 moves to the carriage-return point and down 1 CP-cell line; labels "MARGIN"

70 returns the pen to the carousel

Notice that a fractional line parameter was specified in line 30 so that the label would not be drawn on the line, as would happen with a line parameter of zero. Also notice that the instruction CP ; in line 60 causes the same moves as the embedded carriage return and line feed in line 50.

The Extra Space Instruction, ES

USES: The ES instruction adjusts the spaces and lines between characters without affecting character size.

SYNTAX: *ES* spaces (, lines) *term*
or
ES *term*

Parameter	Format	Range	Default
spaces	decimal	-2^{23} to $2^{23}-1$	0
lines	decimal	-2^{23} to $2^{23}-1$	0

EXPLANATION: An ES instruction without parameters (ES ;) is equivalent to ES 0 , 0 ;, and defaults spaces and lines between characters to the dimensions of the character plot cell, as set by the most recent SI or SR instruction. When specified, the parameters are interpreted as follows. (For more information on spaces, lines, and the character plot cell, refer to Adjusting Character Size, Spacing, and Position earlier in this chapter.)

1. Spaces. The spaces parameter is in terms of the current character plot cell (or average character plot cell, for variable-space fonts). A parameter of ±1 adds or subtracts one full cell space to the current spacing between characters. To draw legible labels, you will probably find that you should not specify extra space greater than 1; you can specify a decimal fraction.

ManualsPlus.com

2. Lines. The lines parameter is in terms of the current character plot cell (or average character plot cell, for variable-space fonts). A parameter of ±1 adds or subtracts one full cell line to the current line spacing between labels. You can specify a decimal fraction.

Positive space and line values specify extra space to be added; as a result, characters will be drawn further apart. Negative values specify extra space to be subtracted; as a result, characters will be drawn closer together, even overlapping.

The following illustration shows fixed-space and variable-space fonts with default, decreased, and increased spacing. Notice that changing spacing on variable-space fonts tends to make them look as though they were fixed-space. (To select variable-space fonts, refer to Chapter 11.)

```
ES;  CAUSES
THIS SPACING.

ES-.1,-.25;  CAUSES
THIS SPACING.

ES.2,.25;  CAUSES
THIS SPACING.
```

Character Set 0
(Fixed Space)

```
ES; CAUSES
THIS SPACING.

ES-.1,-.25; CAUSES
THIS SPACING.

ES.2,.25; CAUSES
THIS SPACING.
```

Character Set 10
(Variable Space)

The ES instruction remains in effect until another ES instruction is executed, or the plotter is initialized or set to default conditions.

The following table summarizes the possible error conditions or unexpected results that you might observe with the ES instruction.

Condition	Error	Plotter Response
no parameters	none	establishes no extra spaces or lines
1 parameter	none	adjusts space, but not line
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction

ManualsPlus.com

The Buffer Label Instruction, BL

USES: The BL instruction stores a label string in the label buffer. Use this instruction when you want to determine the space requirements of a label prior to drawing it.

SYNTAX: *BL* c ... c *term*
or
BL *term*
(where *term* is the label terminator defined by the DT instruction)

Parameter	Format	Range	Default
c ... c (up to 150 characters are buffered)	label	any character	none

EXPLANATION: All printing characters following the BL instruction are stored in the label buffer. The capacity of the label buffer limits the length of the label string to 150 characters, including control characters and the label terminator.

NOTE: To terminate the BL instruction, you must use the label terminator instead of the usual HP-GL terminator. The default label terminator is the nonprinting end-of-text character (**ETX**, decimal code 3). You can change the label terminator with the define terminator instruction, DT, described earlier in this chapter. ■

The BL instruction clears the label buffer, then stores the first 150 characters in the label buffer. The current BL contents remain in the buffer until another BL or LB instruction overwrites the buffer, a BL or LB instruction without parameters or any CM instruction clears the buffer, or the plotter is initialized or set to default conditions.

The BL instruction does not draw labels. Once the label string is stored in the label buffer, you can draw it by executing the PB instruction. You can also use the BL instruction in conjunction with the OL instruction to obtain information about the size of the label string prior to drawing it. Refer to The Output Label Length Instruction, OL, and the Print Buffered Label Instruction, PB, next in this chapter.

NOTE: The label buffer is independent of the buffers in the configurable graphics memory. You cannot change the size of this buffer. ■

The following table summarizes the possible error conditions or unexpected results that you might observe with the BL instruction.

ManualsPlus.com

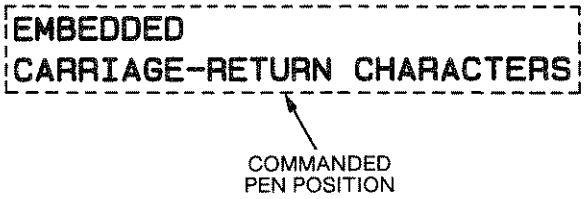
Condition	Error	Plotter Response
no parameter	none	clears label buffer
more than 150 characters	none	stores 150 characters; ignores the rest

Example — Positioning Buffered Labels with the LO Instruction

Buffered labels, like labels drawn with the LB instruction, are subject to positioning and control characters. However, when embedded carriage-returns are used, an LO instruction positions a buffered label differently from an LB label. Each line is left-justified and the entire label is positioned (using the label origin in effect) as though it were enclosed in a rectangular box the length of the longest text line.

This example shows how the LO instruction positions a buffered label. Compare this to the example given previously under the LO instruction, where each line of the label is centered.

```
"IN;SP1;PA6000,4000;LO14;"
"BLEMBEDDED"+CHR$(13)+CHR$(10)+"CARRIAGE-RETURN
  CHARACTERS"+CHR$(3)
"PB;SPO;"
```



The Print Buffered Label Instruction, PB

USES: The PB instruction plots the contents of the label buffer. Use this instruction to repeatedly plot the contents of the label buffer.

SYNTAX: *PB term*

EXPLANATION: The PB instruction plots the contents of the label buffer at the current pen position using the parameters of any ES, DI, DR, LO, SI, SL, and SR instructions in effect when the PB is executed. You can change spacing, direction, position, slant, and size between repeated printings of a label.

However, the setting of the label origin (LO instruction) can limit the use of PB. If you are using label origins 1 through 3 or 11 through 13, you can

ManualsPlus.com

repeat the last label any number of times. If you are using other label origins *and* the label contains an embedded carriage return (decimal code 13), only the part of the label following the carriage return will be repeated. To print the entire label, use the BL instruction instead of the LB instruction prior to printing the label with PB.

The label buffer contains the most recent string of label characters received in a BL or LB instruction and holds a maximum of 150 characters. The label buffer is not cleared by any PB instruction. The buffer contents remain until a CM instruction or an LB or BL instruction without parameters is executed, or the plotter is initialized or set to default conditions.

For examples using the PB instruction, refer to The Label Instruction, LB, The Buffer Label Instruction, BL, and The Output Label Length Instruction, OL, all in this chapter.

The following table summarizes the possible error conditions or unexpected results that you might observe with the PB instruction.

Condition	Error	Plotter Response
no parameter	none	plots current contents of label buffer
1 or more parameters	2	plots current contents of label buffer
no label in buffer	none	ignores instruction

The Output Label Length Instruction, OL

USES: The OL instruction outputs information on the contents of the label buffer. Use this instruction along with the BL instruction to determine the space needed for the buffered label prior to drawing it.

SYNTAX: *OL term*

RESPONSE: length, characters, line feeds [TERM]

EXPLANATION: After an OL instruction is received, the plotter outputs information concerning the buffered label as one decimal and two integers in ASCII, separated by commas and followed by the output terminator. If the buffer is empty, three zeros are output.

1. Length — contains the length of the longest line in the buffered label in character plot cell spaces. The length is output with 4 decimal places.
2. Characters — contains an integer that represents the number of printing characters and spaces in the longest line of the buffered

ManualsPlus.com

label. A backspace counts as −1; a character with automatic backspace (for example, some accent characters) counts as zero.

3. Line Feeds — contains an integer representing the net number of line feeds that will occur when the buffered label is drawn. An inverse line feed (VT, CHR$(11)) counts as −1, and a line feed (LF, CHR$(10)) counts as +1. If the buffered label contains the same number of both types of line feeds, zero is output.

The OL instruction is normally used with the BL instruction, but outputs information on the current contents of the buffer whether it was filled with the LB or BL instruction.

After you send the OL instruction, you should read the output response. Refer to Hints for Obtaining Plotter Output Responses in Chapter 13.

The following table summarizes the possible error conditions or unexpected results that you might observe with the OL instruction.

Condition	Error	Plotter Response
no parameter	none	outputs label information
1 or more parameters	2	outputs label information

Example — Underlining a Label Using Information from OL

This example uses the OL instruction to determine the correct length of a label so that the entire label can be underlined. You must read the OL values into the computer in order to use them (line 50, INPUT #1).

Underline this label.

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;PA5000,3000;"
30  PRINT #1, "BLUnderline this label."+CHR$(13)+
              CHR$(3)
40  PRINT #1, "OL;"
50  INPUT #1, A,B,C
60  PRINT #1, "PB;"
70  PRINT #1, "CP0,-.25;PD;CP";A;",0;"
80  PRINT #1, "SP0;"
90  END
```

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; moves to the starting label position

ManualsPlus.com

30 places a label in the buffer (the label ends with a carriage return so that the current pen position will return to the starting point of the label)

40 causes the plotter to output label information

50 system-dependent BASIC statement that causes the computer to read in the response parameters of the OL instruction (refer to Chapter 13 for more information on reading the output response)

60 draws the buffered label

70 moves the pen down one-fourth of a line feed; lowers the pen; draws a line to the end of the label by using the length output by the OL instruction

80 returns the pen to the carousel

Parameter Interaction in Labeling Instructions

There are three interacting factors that affect the direction and mirroring of labels: the label direction as specified by DI or DR instructions, the sign of the parameters for the size instructions SI or SR, and the relative positions of P1 and P2. These interactions are complex. This section considers the four possible combinations of DI, DR, SI, and SR and illustrates the effects of various parameters and settings of P1 and P2 on labels.

The labels used in the illustrations are the instructions that cause the direction, size, and mirroring of the label. All descriptions are in terms of the standard X,Y coordinate system; quadrant numbers refer to those shown in the first example under The Absolute Direction Instruction, DI. An arrow is shown for each label; this arrow is the baseline along which labeling occurs, and shows the left-to-right direction that is the standard direction of a label without mirroring. The same P1/P2 area, that area set by default P1 and P2 for A4/A-size paper, is always used. During the course of the illustrations, P1 and P2 are assigned to opposite corners of this rectangle in all possible ways. All illustrations assume that scaling is **not** in effect.

Using DI and SI Together

When the DI and SI instructions are used together, the DI instruction establishes the label's direction and the SI instruction establishes its size. The direction serves as the axis along and about which labels are mirrored. (To mirror labels in this case, use negative SI parameters.) Positions of P1 and P2 do not affect the labels.

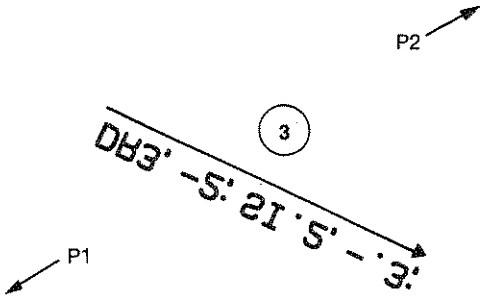
ManualsPlus.com

Two examples of mirrored labels are shown below. In illustration 1, the DI parameters 3,2 place the directional line in the first quadrant. The negative width parameter of the SI instruction mirrors the label in the right-to-left direction. In illustration 2, the DI parameters 3,−2 place the directional line in the fourth quadrant. The negative height parameter of the SI instruction mirrors the label from top to bottom.

Using DR and SI Together

When the DR and SI instructions are used together, the label size is determined by the SI instruction and does not change with changes in the settings of P1 and P2. However, changes in the settings of P1 and P2 will affect the label direction. The plotter multiplies the algebraic differences ($P2_X - P1_X$) and ($P2_Y - P1_Y$) by the run and rise parameters of the DR instruction. The resulting parameters, when applied to the standard coordinate system, determine the label baseline. Mirroring about this baseline is determined by the signs of the SI parameters.

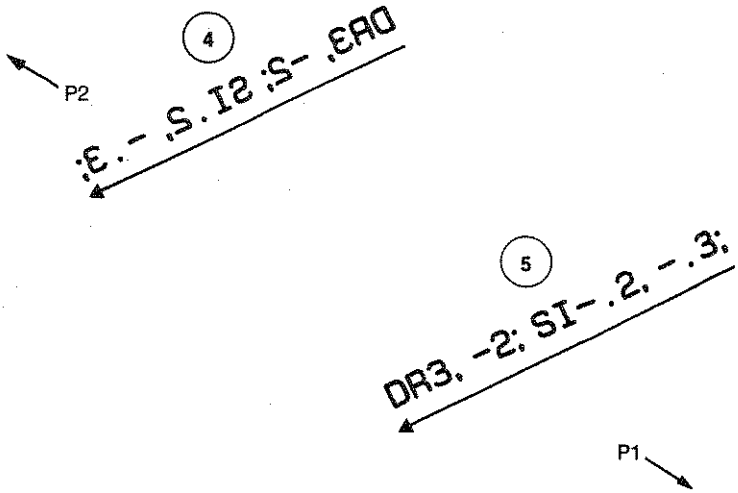
In illustration 3, P1 and P2 are at their default settings for A4/A-size paper, so the algebraic differences ($P2_X - P1_X$) and ($P2_Y - P1_Y$) are both positive. The DR parameters 3,−2 are used as is and establish the directional line in the fourth quadrant. The negative SI height parameter mirrors the label from top to bottom.



In illustrations 4 and 5, P1 is moved to the lower-right corner and P2 is moved to the upper-left corner. Now ($P2_X - P1_X$) is negative. The DR instruction as given is DR 3,−2; the run parameter of the DR instruction is multiplied by −1 and the effective DR instruction becomes DR −3,−2, placing the directional line in the third quadrant. The negative SI height

ManualsPlus.com

parameter mirrors the label from top to bottom. In illustration 5, both SI parameters are negative and the label is mirrored in both directions, making it appear upright.

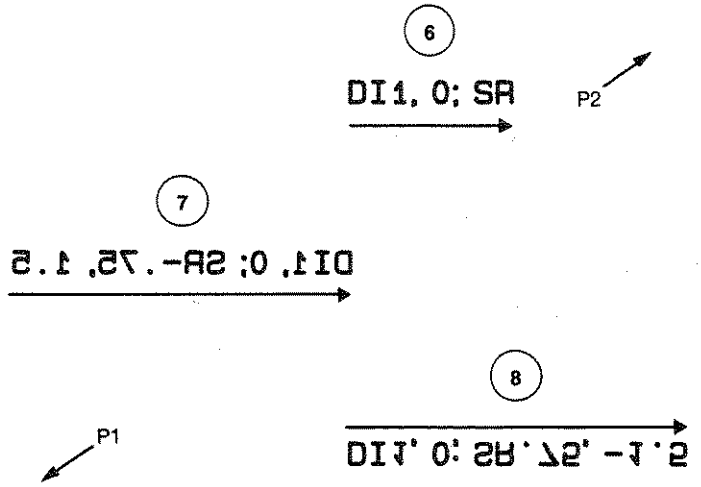


Using DI and SR Together

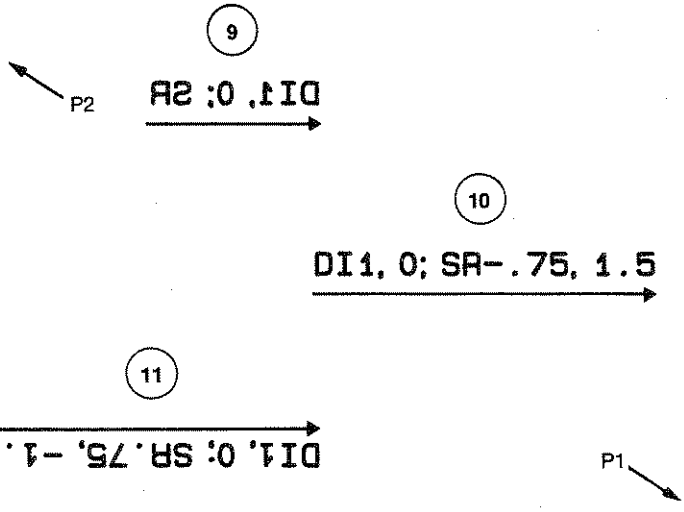
When the DI instruction is used with SR, only the DI instruction affects the directional baseline of labels; changes in the relative positions of P1 and P2 do not affect the baseline. Mirroring about this baseline will occur when either a negative SR width or height parameter with a positive difference ($P2_X - P1_X$) **or** ($P2_Y - P1_Y$), or a positive SR parameter and a negative difference are present. If respective parameters and differences are both positive or both negative, no mirroring will occur.

Label direction is horizontal for all illustrations in this section. The first three illustrations are drawn with P1 and P2 at their power-on settings. In illustration 6, the SR; instruction is the same as SR.75,1.5. Since the parameters are positive, there is no mirroring. In illustration 7, the negative width parameter causes mirroring from right to left. In illustration 8, the negative height parameter causes mirroring from top to bottom.

ManualsPlus.com

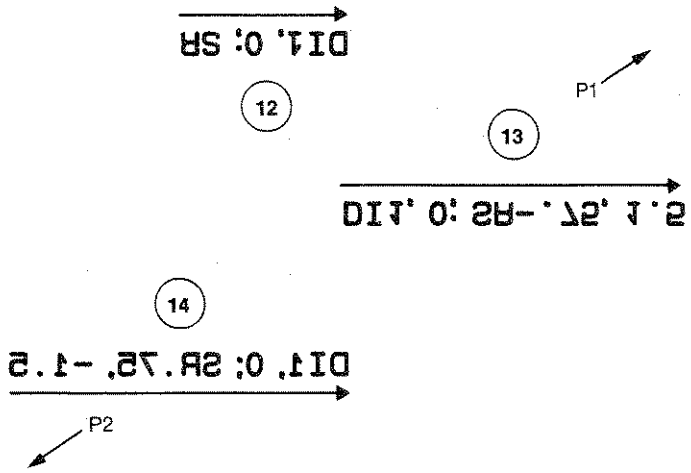


In the next three illustrations, P1 and P2 have been changed so that P1 is in the lower-right corner and P2 is in the upper-left corner. Hence ($P2_X - P1_X$) is negative and anything with a positive SR width parameter is mirrored from right to left (illustrations 9 and 11). The effect of the negative width parameter in illustration 10 is cancelled by the negative difference ($P2_X - P1_X$).



In the next illustrations, P1 and P2 have both been flipped so that P1 is in the upper-right corner and P2 is in the lower-left corner. Now any positive parameter causes mirroring and any negative parameter cancels mirroring. This can be seen in illustrations 12, 13, and 14.

ManualsPlus.com



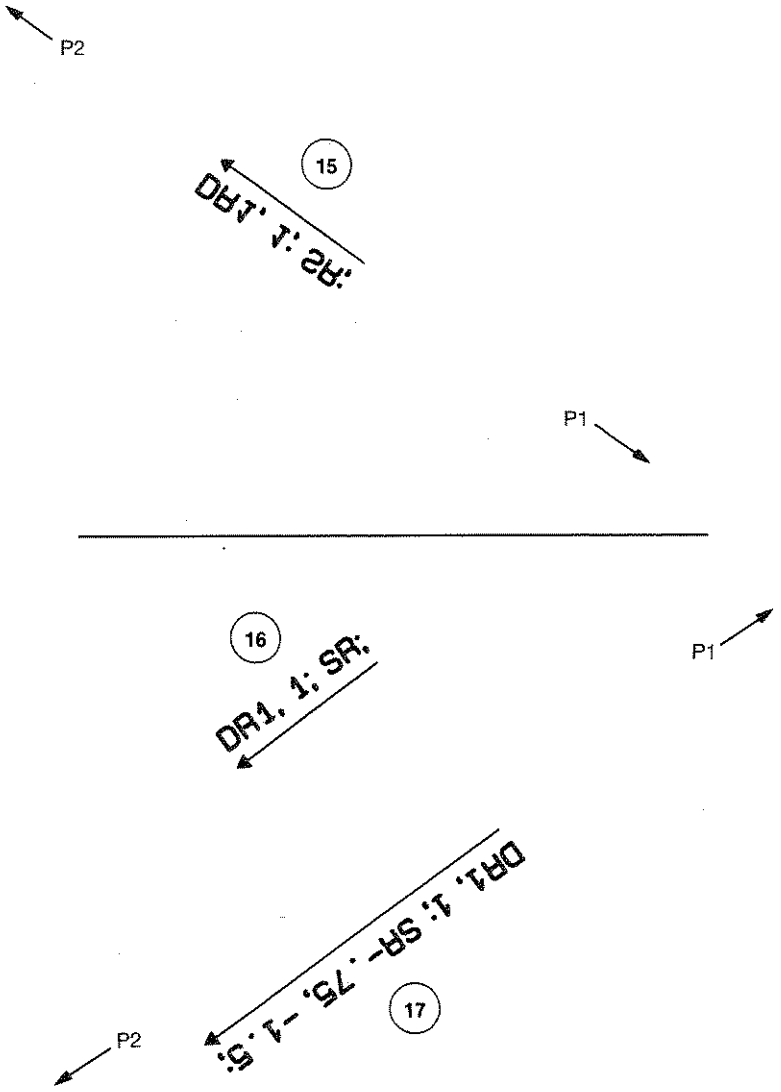
Using DR and SR Together

When the DR and SR instructions are used together, interactions are the most complex. Using only standard settings of P1 and P2, where P1 is in the lower-left corner and P2 is in the upper-right corner, will make it easier for you to establish the direction and mirroring of labels you desire. DR parameters interact with the algebraic differences ($P2_X - P1_X$) and ($P2_Y - P1_Y$) to establish label direction, and SR parameters interact with these differences to create mirroring. Signs of both parameters and P2/P1 differences are important. A negative sign in either the parameter or the difference will affect both DR and SR instructions. Having both parameter and P2/P1 difference either positive or negative will cause standard direction or no mirroring.

The following examples show the most complex cases, with P1 and P2 in nonstandard locations. Label 15 is drawn with the instructions DR 1 , 1 ; SR in effect, P1 in the lower-right corner and P2 in the upper-left corner. The label baseline is in the second quadrant, not the first, because ($P2_X - P1_X$) is negative and the DR run parameter is positive.

Labels 16 and 17 are drawn with the same instructions, but with P1 in the upper-right corner and P2 in the lower-left corner. The label direction baseline is in the third quadrant because both ($P2_X - P1_X$) and ($P2_Y - P1_Y$) are negative. Label 16 is mirrored in both directions. (Rotate the manual so that the arrow points to +45 degrees to see this more clearly.) Label 17 is not mirrored because both parameters and differences are negative. (Again, this may be easier to see if you rotate the manual.)

ManualsPlus.com



ManualsPlus.com

ManualsPlus.com

Chapter 8
Putting the Instructions to Work

What You'll Learn in This Chapter

In this chapter you'll learn how to put instructions together to develop a plot. Previous programs have purposely been kept to a less advanced level in order to clearly demonstrate the instruction usage. The following examples are designed to show you how to integrate many instructions into a complete program, how data might be handled, and how subroutines might be used to program a task that would be common to many plots and used in several programs.

- The first program draws a line chart, one of the most common types of plots. While this line chart shows sales data, line charts can be used to plot almost any kind of data — factory output, sales volume, data from laboratory experiments, population trends, etc. The concepts of plotting and labeling demonstrated here can be used in almost any application.
- The second program draws a stacked bar chart. The finished chart, entitled "Sales Volume by Region," shows the yearly sales for various regions. The sales data are stacked in bars for each region, and are differentiated by solid fill, cross-hatching, and parallel hatching. This program demonstrates the use of the FT, RA, and EA instructions to define fill types, and to fill and outline rectangles.
- The third program draws a pie chart. The finished pie chart, entitled "Sales Dollar Distribution," shows the sales dollar distribution among four groups. The sales data are represented by four wedges, which are differentiated by solid fill, cross-hatching, and parallel hatching. This program demonstrates the use of the FT, WG, and EW instructions to define fill types, and to fill and outline pie wedges.

The first program (the line chart), is explained in full detail, and organized to show you how to develop a program. The second two programs are explained more briefly, because the concepts of developing these programs are similar to developing the line chart.

ManualsPlus.com

A Reminder about HP-GL Syntax

HP-GL syntax allows a variety of separators and terminators. However, good programming techniques dictate the use of a consistent style. The style used in these examples should be compatible with most computers and interfaces. In applications where compatibility with older HP plotters is important, you should always use a semicolon or a line feed as the terminator and separate parameters with commas. With RS-232-C interfaces, use a semicolon; line feeds are not recognized as terminators.

A Reminder about BASIC

These programs are written in Microsoft® BASIC. They use techniques such as FOR... NEXT loops and subroutines to read data and draw the plots. Remember to check your computer documentation for the correct methods of implementing these techniques.

Line Chart

For this line chart, you will scale, draw, and label an X- and Y-axis in user units and plot 1985 sales by sales region. You will use a different line type for each sales region and place a legend on the chart.

The following paragraphs develop segments of the program in a logical sequence. The complete plot and program are shown later, in the section titled Program Listing.

Setup and Scaling

For emphasis and readability, you should draw the data curves and title with wide pens. Narrow pens are usually sufficient for axes and labels. For this line chart, the suggested pen order for the carousel is:

1 = black, P.3	5 = blue, P.7
2 = black, P.7	6 = green, P.7
3 = purple, P.7	7 = (unused)
4 = red, P.7	8 = (unused)

If you do not have any wide pens (P.7), use narrow pens (P.3). You may purchase wide pens from Hewlett-Packard; part numbers are listed in the Operation and Interconnection Manual under Accessories Available.

Now begin your program by setting the plotter to known conditions, cancelling any parameters which may have been set during a previous plot. The IN or DF instruction may be used. IN is used here to be sure *all* conditions (such as P1/P2 settings) are set to a default state so that this example will plot exactly as intended. You might find it more convenient to use DF in your programs instead, so that the plotter operator can set P1, P2, pen speed, and other functions from the front panel before running a program.

ManualsPlus.com

Next select a pen (SP 1) and establish the scaling for this plot. The parameters of the IP instruction determine the location of the scaling points, P1 and P2. The locations of these points were chosen to provide a convenient area for the scale, which is assigned in the scaling statement SC 1 , 12 , 0 , 150 ;. Since this chart shows one year's sales by month, the X-axis (commonly representing time) is scaled from 1 to 12. The Y-axis is scaled in thousands from 0 to 150 so that all sales data will fall inside the scaled area. Labels and titles will be placed outside this area.

You will either need to know the range of your data or be willing to go through some trial plots with different scales to determine what your scale statement should be. This chart is scaled from 0 to 150, not 0 to 150 000 — the actual range of sales dollars. The shorter labels along the axis are easier to read and prevent the chart from becoming cluttered; thus, the chart will be easier to interpret. Thousands or millions of dollars are common scales.

Having established the scale, draw a frame for the data area. This is done by moving to the point 1 , 0 and specifying the diagonally opposite corner in the EA instruction.

The first three program lines with HP-GL instructions are:

```
20    PRINT #1, "IN;SP1;IP1250,750,9250,6250;"
30    PRINT #1, "SC1,12,0,150;"
40    PRINT #1, "PU1,0;EA12,150;"
```

NOTE: If compatibility with older HP plotters is desired, PA should be used to begin plotting, and raising and lowering the pen should be controlled with separate PU and PD instructions. ■

The Axes and Their Labels

You are now ready to draw and label the axes. The label size is set by the absolute size instruction SI .2 , .3 ;. This creates characters which are slightly larger than characters of the default character size specified by the IN instruction. The tick length is established by the tick length instruction TL 1.5 , 0 ;. The resulting ticks will be 1.5% of the horizontal or vertical distances between the scaling points. No negative portion of the tick will be drawn; ticks will be entirely above the X-axis and to the right of the Y-axis.

Axes are commonly drawn using a loop; this program uses FOR . . . NEXT loops. First, draw the X-axis. Let X range from 1 to 12, representing the 12 months for which you have data. The loop will do four things: move to the integer location on the axis, draw a tick mark, establish the label origin, and draw the label. Note that the X-parameter of the plot instruction is a

ManualsPlus.com

variable. You will need to know how to send a variable between strings of fixed characters. The method differs from computer to computer; consult your computer's documentation and Plotting with Variables in Chapter 4 of this manual. If you are using the HP-IB interface, refer also to Sending and Receiving Data in Chapter 15. The XT instruction draws a tick, whether the pen is up or down. The pen is up here to avoid unwanted lines between the ticks, labels, and axis.

In order to use the looping techniques for labeling axes, you must place the labels in a data statement. (At some point, you might want to access data for the latest 12 months. If your data were stored together with a date code, you could use a similar technique to read the labels and data from some file and properly label your chart for the data you were then plotting.) You then access the labels with a string variable in the LB instruction. Refer to Labeling with Variables in Chapter 7 for hints on sending variables in labels.

To position the labels, use the LO instruction. The base of the tick mark becomes the current pen position after the tick is drawn. Specifying LO 16 causes the label to be centered under the tick and offset down slightly so that the label can be easily read. For the X-axis title (calendar month), the pen is first moved to the center of the graph below the X-axis. Then, LO 5 centers the label over this position.

The loop to draw the axis, and the instructions and data statements to label and title the X-axis are:

```
50   PRINT #1, "SI.2,.3;TL1.5,0;"
60   FOR X=1 TO 12
70     PRINT #1, "PA";X;",0;XT;"
80     READ A$
90     PRINT #1, "LO16;LB";A$;+CHR$(3)
100  NEXT X
110  PRINT #1, "PA6.5,-10;LO5;LBCalendar Month"+
                  CHR$(3)


380  DATA "J","F","M","A","M","J"
390  DATA "J","A","S","O","N","D"
```

The Y-axis is created in a similar manner, except the loop's index is used for the label value (0–150), and the LO 8 instruction is used to place labels to the left of the tick marks. The Y-axis title is centered above the axis.

Following the axis loop are the instructions that label the regions for the legend. The legend is drawn now while the label size is small and the narrow pen is in the pen holder. The lines for the legend will be

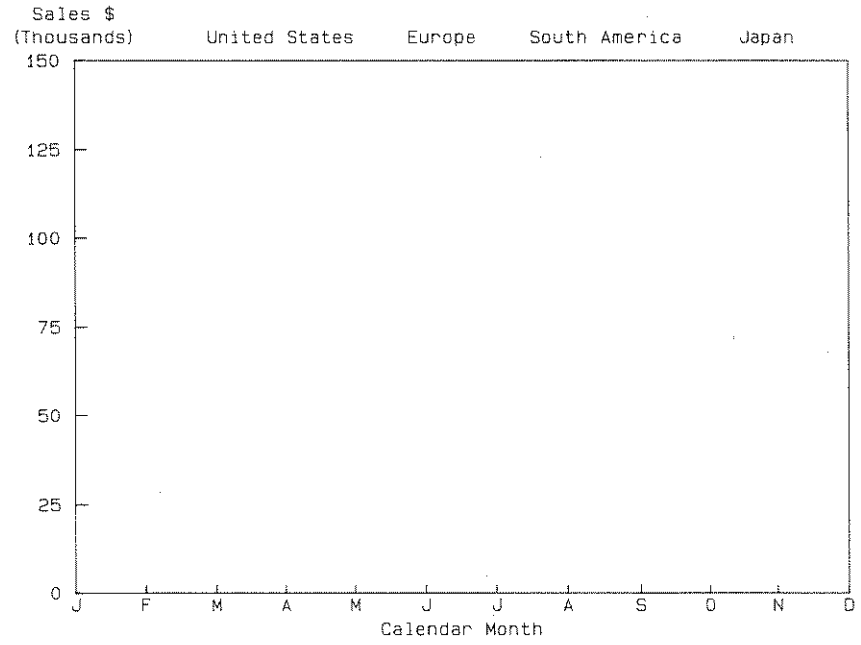
ManualsPlus.com

drawn later as each line of data is plotted. The labels for the legend are positioned by using LO 16 and PA instructions to center them below the areas where the lines will be drawn.

The lines which draw the Y-axis, label it, and draw the legend labels follow.

```
120   FOR Y=0 TO 150 STEP 25
130     PRINT #1, "PA1,",Y,"YT;"
140     PRINT #1, "LO8;LB";Y;+CHR$(3)
150   NEXT Y
160   PRINT #1, "PA1,160;LO14;LBSales $"+CHR$(13)+
                  CHR$(3)+"LO16;"
170   PRINT #1, "LB(Thousands)"+CHR$(3)
180   PRINT #1, "PA3.95,160;LBUnited States"+CHR$(3)
190   PRINT #1, "PA6.25,160;LBEurope"+CHR$(3)
200   PRINT #1, "PA8.55,160;LBSouth America"+CHR$(3)
210   PRINT #1, "PA10.85,160;LBJapan"+CHR$(3)
```

Here's what the chart looks like so far.



The most important parts of the chart are drawn with a wide pen. Move to the top center, increase the character size, and label the chart title.

The program lines to title the chart using a wide pen are:

```
220   PRINT #1, "SP2;PA6,175;SI.4,.6;LO4;"
230   PRINT #1, "LB1985 Sales by Region"+CHR$(3)
```

ManualsPlus.com

Plotting the Data

You are now ready to draw lines. Each of the four data lines on this chart is drawn using a different technique. The first two lines are drawn by plot instructions with parameters included when the program was written. Hence, if the data changes, it will be necessary to change the plot instructions in the program.

The first line (the bottom-most line on the graph) is drawn with pen 3 using a dashed line type. After drawing the line, the pen moves to the legend area below the chart title and draws a short line. The PU instruction causes the line type pattern to begin again at the begining of this line.

The second line is also plotted using plot instructions with fixed parameters. The line type used consists of long and short dashes; the line is drawn with pen 4. After the data points are plotted, the corresponding line is drawn in the legend.

The program lines which plot the two lower lines and the corresponding legend lines are:

```
240   PRINT #1, "SP3;LT3,6;PA1,23;PD2,25,3,18,4,22;"
250   PRINT #1, "PD5,23,6,27,7,27,8,25,9,24,10,28;"
260   PRINT #1, "PD11,27,12,27;PU7.8,165;PD9.3,165;"
265   PRINT #1, "PU;SP4;LT6,8;PA1,45;"
270   PRINT #1, "PD;PA2,50,3,52,4,53;PD5,52,6,51;"
280   PRINT #1, "PD7,55,8,56,9,56,10,58,11,58;"
290   PRINT #1, "PD12,60;PU10.1,165;PD11.6,165;"
```

The third line is plotted from data read by the program using a FOR...NEXT loop and a READ statement. This technique would be used to plot a chart that will be replotted often with new data. If the necessary file statements were added, the data could be on a tape or disk file instead of in a DATA statement as shown here. The line type for this line is the default solid line, reverted to by the LT instruction with no parameters. Since this program uses variables as plot parameters, be sure they are sent to the plotter with a valid separator between them. Computers often send a leading and/or trailing blank, or allow for a sign space before numeric variables. The HP 7550 will treat a blank or a comma as a separator between numeric parameters. Know your computer before sending variables with plot instructions. As with the two previously drawn lines, after the line is plotted, the corresponding line is placed in the legend.

ManualsPlus.com

The loop to plot this third line using pen 5, and the instructions to place a line in the legend are:

```
300   PRINT #1, "PU;LT;SP5;"
310   FOR X=1 TO 12
320     READ Y
330     PRINT #1, "PA";X;",";Y;";PD;"
340   NEXT X
350   PRINT #1, "PU5.5,165;PD7.0,165;PU;"

400   DATA 55,60,63,62,59,54,50,46,47,49,53,58
```

The last line is drawn using a subroutine. The subroutine is designed to read data that have been stored with a third value for pen control. This third value controls a branch to two different plot instructions, one with the pen up and the other with the pen down. In this program, a 0 for the pen control parameter results in a pen up move, a 1 causes plotting with the pen down, and 3 signifies the end of the data. The last line is drawn with pen 6. The legend line is drawn at the end of the subroutine, completing the chart.

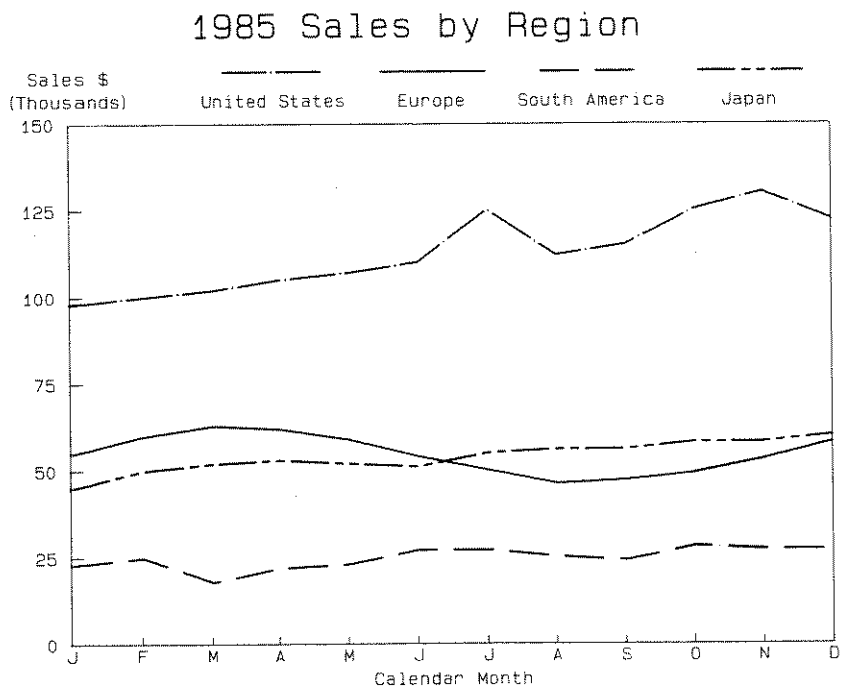
The program lines to change pens and line type, and the subroutine itself are listed here.

```
360   PRINT #1, "SP6;LT4,6;"
370   GOSUB 1000

1000 ' Plotting subroutine through line 1100
1010 READ X,Y,P
1020 IF P=1 THEN PRINT #1, "PD";X;",";Y;";"
1030 IF P=0 THEN PRINT #1, "PU";X;",";Y;";"
1040 IF P=3 THEN 1090
1050 DATA 1,98,0,2,100,1,3,102,1,4,105,1,5,107,1
1060 DATA 6,110,1,7,125,1,8,112,1,9,115,1
1070 DATA 10,125,1,11,130,1,12,122,1,0,0,3
1080 GOTO 1010
1090 PRINT #1, "LT4,6;PU3.2,165;PD4.7,165;SP0;"
1100 RETURN
```

Program Listing

A reduced version of the plot is shown next, followed by a complete listing of the program. This listing contains all of the BASIC statements necessary to have this program run on an HP Touchscreen (150) computer. You might need to make changes for your computer's BASIC, or you can use some other programming language and send the strings of HP-GL instructions using your language's output statements and looping techniques.

ManualsPlus.com




```
10    'Insert configuration statement here
20    PRINT #1, "IN;SP1;IP1250,750,9250,6250;"
30    PRINT #1, "SC1,12,0,150;"
40    PRINT #1, "PU1,0;EA12,150;"
50    PRINT #1, "SI.2,.3;TL1.5,0;"
60    FOR X=1 TO 12
70      PRINT #1, "PA";X;",0;XT;"
80      READ A$
90      PRINT #1, "LO16;LB";A$;+CHR$(3)
100   NEXT X
110   PRINT #1, "PA6.5,-10;LO5;LBCalendar Month"+
                  CHR$(3)
120   FOR Y=0 TO 150 STEP 25
130     PRINT #1, "PA1,";Y;";YT;"
140     PRINT #1, "LO8;LB";Y;+CHR$(3)
150   NEXT Y
160   PRINT #1, "PA1,160;LO14;LBSales $"+CHR$(13)+
                  CHR$(3)+"LO16;"
170   PRINT #1, "LB(Thousands)"+CHR$(3)
180   PRINT #1, "PA3.95,160;LBUnited States"+CHR$(3)
190   PRINT #1, "PA6.25,160;LBEurope"+CHR$(3)
200   PRINT #1, "PA8.55,160;LBSouth America"+CHR$(3)
210   PRINT #1, "PA10.85,160;LBJapan"+CHR$(3)
220   PRINT #1, "SP2;PA6,175;SI.4,.6;LO4;"
```

(Program listing continued)

ManualsPlus.com

```
230  PRINT #1, "LB1985 Sales by Region"+CHR$(3)
240  PRINT #1, "SP3;LT3,6;PA1,23;PD2,25,3,18,4,22;"
250  PRINT #1, "PD5,23,6,27,7,27,8,25,9,24,10,28;"
260  PRINT #1, "PD11,27,12,27;PU7.8,165;PD9.3,165;"
265  PRINT #1, "PU;SP4;LT6,8;PA1,45;"
270  PRINT #1, "PD;PA2,50,3,52,4,53;PD5,52,6,51;"
280  PRINT #1, "PD7,55,8,56,9,56,10,58,11,58;"
290  PRINT #1, "PD12,60;PU10.1,165;PD11.6,165;"
300  PRINT #1, "PU;LT;SP5;"
310  FOR X=1 TO 12
320    READ Y
330    PRINT #1, "PA";X;",";Y;";PD;"
340  NEXT X
350  PRINT #1, "PU5.5,165;PD7.0,165;PU;"
360  PRINT #1, "SP6;LT4,6;"
370  GOSUB 1000
380  DATA "J","F","M","A","M","J"
390  DATA "J","A","S","O","N","D"
400  DATA 55,60,63,62,59,54,50,46,47,49,53,58
410  END
1000 ' Plotting subroutine through line 1100
1010 READ X,Y,P
1020 IF P=1 THEN PRINT #1, "PD";X;",";Y;";"
1030 IF P=0 THEN PRINT #1, "PU";X;",";Y;";"
1040 IF P=3 THEN 1090
1050 DATA 1,98,0,2,100,1,3,102,1,4,105,1,5,107,1
1060 DATA 6,110,1,7,125,1,8,112,1,9,115,1
1070 DATA 10,125,1,11,130,1,12,122,1,0,0,3
1080 GOTO 1010
1090 PRINT #1, "LT4,6;PU3.2,165;PD4.7,165;SP0;"
1100 RETURN
```

Bar Charts and Pie Charts

Filling and Hatching

Two kinds of area fill are commonly used in bar and pie charts; solid fill and hatching. Solid fill totally covers the area with color, whereas hatching fills the area with evenly spaced parallel lines. If there are two sets of parallel lines at 90-degree angles to each other, the fill is called "cross-hatching." To avoid optical illusions that affect one's perception of the size of a bar or pie segment, it is best to place hatching or cross-hatching at a 45-degree angle from vertical.

NOTE: For both the bar and pie charts, use the same pen order in the carousel that was recommended for the line chart. ■

ManualsPlus.com

Producing a Bar Chart

Overview

This program plots a stacked bar chart, titled "Sales Volume by Region." A stacked bar chart is appropriate when you want to compare parts of an element over time. In this case, the sales volumes of three different regions are compared over a period of three years. For ease of understanding, stacked bar charts should not contain more than six sets of stacks; each stack should be limited to three or four segments.

The following paragraphs describe the program. The line numbers refer to the complete program listing, which is presented after all of the descriptions.

Setting Up the Data (Lines 10-130)

After dimensioning arrays and declaring the label and bar segment data, a loop is used to read the data into the arrays. The arrays will be used later to plot the data. This method is particularly convenient when you anticipate having new data at a future date, because you will only need to change the first few lines of the program to plot a new chart.

Scaling the Axes (Lines 140-160)

The X-axis is scaled by years, from 1982 to 1986. The Y-axis is scaled from 0 to 500 to represent sales in thousands of dollars.

Plotting the Title (Lines 170-200)

The title and axes are drawn with a wide pen for emphasis. The title is drawn first, with characters that are a little more than twice the default size.

Labeling the Axes (Lines 210-320)

The bars on the X-axis are labeled without tick marks, using a narrow pen and characters that are slightly larger than default size. The Y-axis is labeled with tick marks and an extra label to show the scaling used (K$).

Labeling the Bar Segments (Lines 330-430)

The data for labeling each bar segment was input previously in lines 60–80. Using this data, each segment label is centered next to the rightmost bar by computing a Y-axis position that is equal to the height of the prior segment plus one-half the height of the current segment.

Filling and Edging Each Segment (Lines 440-680)

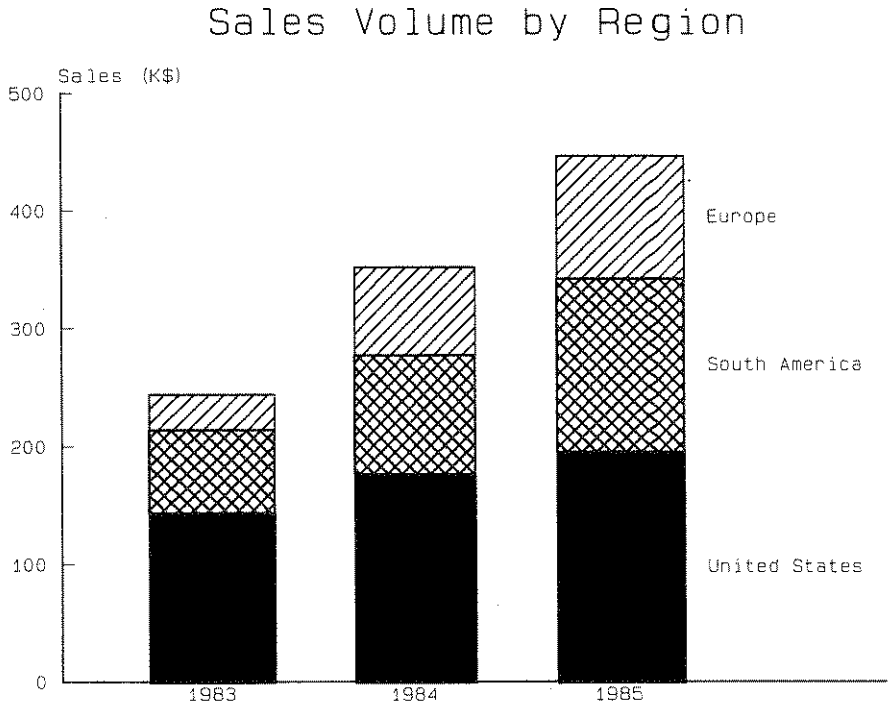
The data for each bar segment was stored previously in a three-by-three array (in lines 120–160). Each array element contains the height of a segment with respect to the Y-axis scaling. This information is used to

ManualsPlus.com

draw the bar segments from bottom to top using the FT, RA, and EA instructions to define, fill, and edge each stacked rectangular segment. Wide pens are used to fill the bars.

Program Listing

The complete bar chart and listing follow. Note that the program includes comments denoted by an apostrophe ('). You do not have to enter them; they are included here to further explain the logic of the program. If you do enter them, check your computer language documentation for the correct comment symbol. In BASIC, comments are often denoted by the statement REM.




```
10  'Insert configuration statement here
20  'Place chart and label data in arrays
30  'with lower bounds of 1
40  OPTION BASE 1
50  DIM L$(3),B(3,3)
60  DATA United States,South America,Europe
70  DATA 144,177,196,71,101,147,30,75,104
80  READ L$(1),L$(2),L$(3)
90  FOR I=1 TO 3
100   FOR J=1 TO 3
110    READ B(I,J)
120   NEXT J
130 NEXT I
```

ManualsPlus.com

```
140 'Initialize;set P1 & P2;scale axes
150 PRINT #1, "IN;IP1000,1000,9000,6750;"
160 PRINT #1, "SC1982,1986,0,500;"
170 'Label title--wide pen;large letters;draw axes
180 PRINT #1, "SP2;PA1984.3,550;SI.4,.6;"
190 PRINT #1, "LO4;LBSales Volume by Region"+CHR$(3)
200 PRINT #1, "PA1982.3,500;PD1982.3,0,1986.3,0;"
210 'Select narrow pen;reset character size;
220 'set tick length;then label X-axis.
230 PRINT #1, "PU;SP1;SI.2,.3;TL1.5,0;"
240 FOR X=1983 TO 1985
250   PRINT #1, "PA";X;",0;LO16;LB";X;+CHR$(3)
260 NEXT X
270 'Add ticks and labels to the Y-axis
280 FOR Y=0 TO 500 STEP 100
290   PRINT #1, "PA1982.3,";Y;";YT;"
300   PRINT #1, "LO8;LB";Y;+CHR$(3)
310 NEXT Y
320 PRINT #1, "PA1982.3,510;LO1;LBSales (K$)"+CHR$(3)
330 'Center segment labels using prior height
340 'plus 1/2 current height.
350 FOR I=1 TO 3
360   Y=0
370   FOR J=1 TO I-1
380     Y=Y+B(J,3)
390   NEXT J
400   Y=Y+B(I,3)/2
410   PRINT #1, "PA1985.4,";Y;";"
420   PRINT #1, "LO12;LB";L$(I);+CHR$(3)
430 NEXT I
440 'Draw and fill using wide pens in 3-5
450 FOR I=1 TO 3
460   PRINT #1, "SP";I+2;";PT.7;"
470   K=1
480   FOR X=1983 TO 1985
490     Y1=0
500 '   Compute Y-axis start point for each segment
510     FOR J=1 TO I-1
520       Y1=Y1+B(J,K)
530     NEXT J
540 '   Compute Y-axis end point for each segment
550     Y2=Y1+B(J,K)
560     K=K+1
```

(Program listing continued)

ManualsPlus.com

```
570 '    Select fill type.
580      IF I=1 THEN PRINT #1, "FT1;"
590      IF I=2 THEN PRINT #1, "FT4,.05,45;"
600      IF I=3 THEN PRINT #1, "FT3,.05,45;"
610 '    Move to start point;fill bar segment.
620      PRINT #1, "PA";X-.3;",";Y1;";"
630      PRINT #1, "RA";X+.3;",";Y2;";"
640 '    Return to start point;outline segment.
650      PRINT #1, "PA";X-.3;",";Y1;";"
660      PRINT #1, "EA";X+.3;",";Y2;";"
670    NEXT X
680 NEXT I
690 'Put the pen away and unload paper.
700 PRINT #1, "SP0;NR;"
710 END
```

Producing a Pie Chart

Overview

This program plots a pie chart, titled "Sales Dollar Distribution." Pie charts easily convey information concerning parts of a whole. In this case, sales dollar distribution is broken into four groups: R&D, Administration, Marketing, and Manufacturing. For ease of understanding, pie charts should not be broken into less than three nor more than six parts.

The following paragraphs describe the program. The line numbers refer to the complete program listing, which is presented after all of the descriptions.

Setting Up the Data (Lines 10-160)

Using the same method as was used in the bar chart, the start angle, mid-point, and stop angle for each segment are placed into arrays. In addition, the pie segment labels are read into arrays.

Scaling the Chart (Lines 170-190)

To center the pie chart, P1 and P2 are repositioned with the IP instruction. The plotting area is scaled so that 0 , 0 is near the center of the page.

Plotting the Title (Lines 200-220)

The title is centered with the LO instruction, the character size is set to 0.4-cm wide and 0.6-cm high with the SI instruction, and a wide pen is selected with the SP instruction.

ManualsPlus.com

Plotting the Labels (Lines 230-400)

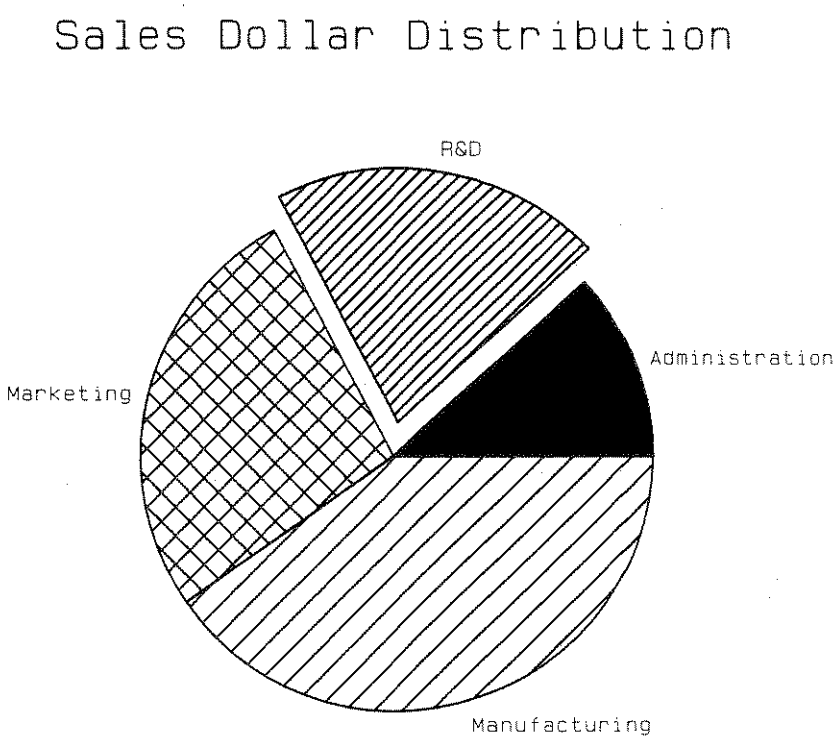
Each segment is labeled using the data previously read in line 160. For readability, the labels are drawn with a narrow pen and with a character size that is smaller than the one used for the title. The labels are centered and offset from the appropriate segment with the LO instruction. The distance used to position the label for segment 2 is increased, since the second segment is offset.

Filling and Edging Each Segment (Lines 410-580)

In this loop, a wide pen and a fill type are selected for each segment with the SP and FT instructions. Lines 440 and 450 adjust the center point for the offset segment. The sweep angle for each segment is then computed (sweep angle = stop angle − start angle), and the segments are filled and outlined with the WG and EP instructions.

Program Listing

The complete pie chart and listing follow. As with the bar chart, comment lines are indicated by an apostrophe ('). You can delete the lines, or use REM instead.



ManualsPlus.com

```
10  'Insert configuration statement here
20  'Place chart and label data in arrays
30  'with lower bounds of 1
40  OPTION BASE 1
50  DIM W$(4),P(4,3)
60  DATA 0,21.5,43,43,80.5,118,118,166.5
70  DATA 215,215,287,360
80  DATA Administration,R&D,Marketing,Manufacturing
90  'Read start, mid-point, and stop angle
100 'for each segment.  Then read labels.
110 FOR I=1 TO 4
120   FOR J=1 TO 3
130     READ P(I,J)
140   NEXT J
150 NEXT I
160 READ W$(1),W$(2),W$(3),W$(4)
170 'Initialize;set P1 & P2;scale axes
180 PRINT #1, "IN;IP2250,500,8250,7100;"
190 PRINT #1, "SC-10,10,-10,12;"
200 'Label title using wide pen, large letters.
210 PRINT #1, "SP2;PA0,12;SI.4,.6;CP-12.34,0;"
220 PRINT #1, "LBSales Dollar Distribution"+CHR$(3)
230 'Label wedges with narrow pen & small letters
240 PRINT #1, "SP1;SI.2,.3;"
250 ' Set PI variable to convert radians to degrees
260 PI=3.141593
270 FOR I=1 TO 4
280 '  Move to center arc.
290    R=8
300    IF I=2 THEN R=9
310    X=R*COS(P(I,2)*(PI/180))
320    Y=R*SIN(P(I,2)*(PI/180))
330    PRINT #1, "PA";X;",";Y;";"
340    Determine label origin, and draw label.
350    L=LEN (W$(I))
360    IF I=1 THEN PRINT #1, "CP0,-.25;"
370    IF I=3 THEN PRINT #1, "CP";-L;",-.25;"
380    IF I=4 THEN PRINT #1, "CP0,-.5;"
390    PRINT #1, "LB";W$(I)+CHR$(3)
400 NEXT I
410 'Draw and fill the wedges using pens 3-6.
420 FOR I=1 TO 4
430   PRINT #1, "SP";I+2;";PT.7;"
440   X=0
450   Y=0
460   IF I=2 THEN X=COS(P(I,2)*(PI/180))
470   IF I=2 THEN Y=SIN(P(I,2)*(PI/180))
480   IF I=1 THEN PRINT #1, "FT1;"
490   IF I=2 THEN PRINT #1, "FT3,.3,45;"
500   IF I=3 THEN PRINT #1, "FT4,.6,45;"
```

(Program listing continued)

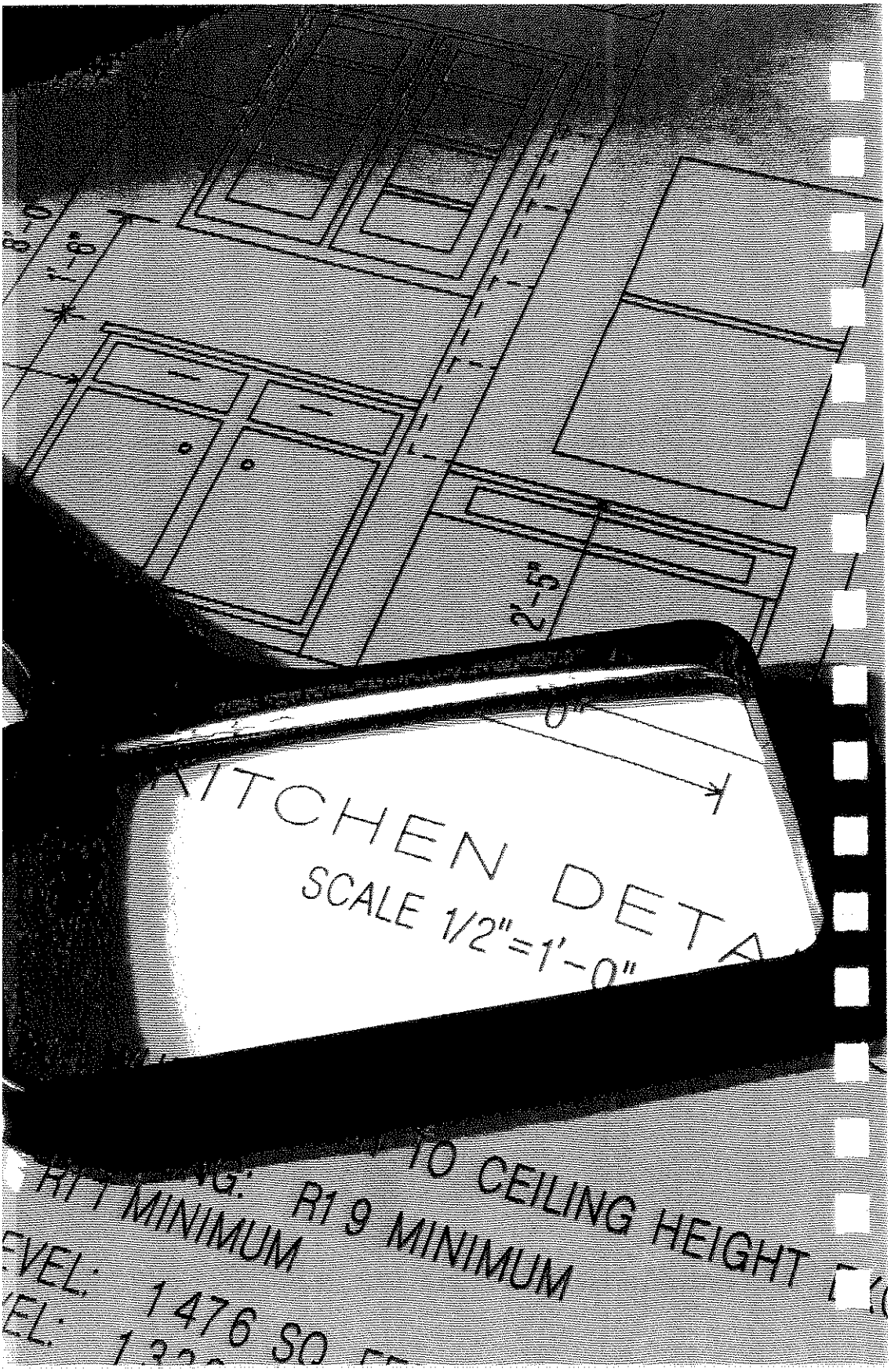
ManualsPlus.com

```
510   IF I=4 THEN PRINT #1, "FT3,.6,45;"
520 ' Compute the sweep angle.
530   S=P(I,3)-P(I,1)
540 ' Fill and outline the wedge.
550   PRINT #1, "PA";X;",";Y;";"
560   PRINT #1, "WG7.5,";P(I,1);",";S;";"
570   PRINT #1, "PA";X;",";Y;";EP;"
580 NEXT I
590 'Put the pen away and unload paper.
600 PRINT #1, "SP0;NR;"
610 END
```

ManualsPlus.com

Notes

ManualsPlus.com



ManualsPlus.com

Chapter 9
Changing the Plotting Area

What You'll Learn in This Chapter

In this chapter you will learn how to make vertical plots by rotating axes, as well as how to restrict plotting to a specific area of the page by defining windows. In addition, you will learn techniques such as proportionally reducing/enlarging your plotting area and creating mirror images.

HP-GL Instructions Covered

RO The Rotate Coordinate System Instruction
IW The Input Window Instruction
OW The Output Window Instruction
OH The Output Hard-Clip Limits Instruction
OP The Output P1 and P2 Instruction

Terms You Should Understand

Hard-Clip Limits — that part of the plotting area beyond which the pen physically cannot move; the mechanical limits of the plotter.

Soft-Clip Limits — that part of the plotting area defined by the IW instruction, beyond which no programmed plotting can occur.

Window — that part of the plotting area in which plotting can occur; nothing can be drawn outside the current window. At power-on, the window is set to the hard-clip limits of the plotter. If soft-clip limits are established, these become the current window.

Clipping — suppressing all points outside of the current window; in effect, restricting plotting to a portion of the plotting area by establishing a window.

ManualsPlus.com

The Rotate Coordinate System Instruction, RO

USES: The RO instruction rotates the plotter's coordinate system (either plotter units or user units) 90 degrees about the plotter-unit coordinate origin. Use this instruction when you want to change the default orientation of your plots. Normally, plots are oriented horizontally on the page; with the RO instruction, you can easily make a vertical plot. (You can also rotate the coordinate system using the front-panel **ROTATE** function key. Refer to the Operation and Interconnection Manual for details.)

SYNTAX: *RO n term*
or
RO term

Parameter	Format	Range	Default
n	integer	0 or 90 degrees	0 degrees

EXPLANATION: Only two parameters are allowed with the RO instruction, as follows.

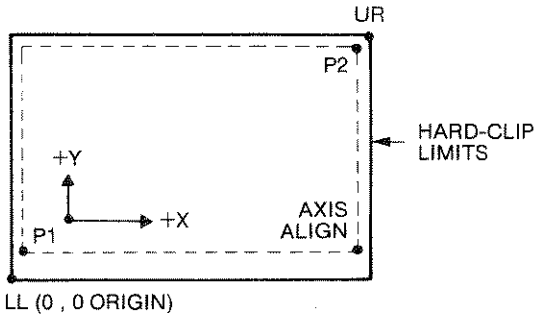
- n=0 Cancel rotation; reestablish default orientation of coordinate system
- n=90 Rotate coordinate system 90 degrees about the plotter-unit origin

An RO instruction without parameters (RO;) is the same as RO 0;. When you issue RO 90;, the plotter's coordinate system is rotated clockwise for A4/A-size paper, and counterclockwise for A3/B-size paper. Thus the orientation for rotated A4/A-size paper is the same as for default A3/B-size paper, and vice versa.

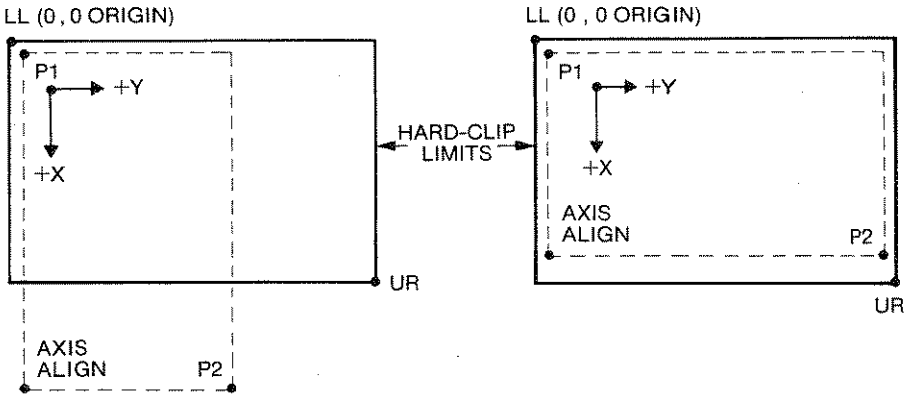
When you use RO 90; to rotate the coordinate system, P1, P2, and the axis-align point rotate with the coordinate system. However, they maintain the same X,Y coordinate values as before the rotation. This means that P2 and the axis-align point are located outside of the hard-clip limits. To reset P1, P2, and the axis-align point so that they are within the new hard-clip limits, issue an IP; instruction after the RO 90; instruction. This effectively switches the X,Y coordinates of each point (e.g., if P1 was 80, 320 before rotation, it will become 320, 80 after rotation).

Each form of rotation is illustrated next. Note that you can also establish or cancel rotation at any time from the front panel. The appropriate function keys to press on the front panel are shown along with the equivalent HP-GL instructions.

ManualsPlus.com



Default Orientation
(RO ; or RO 0 ; or **ROTATE-0**)

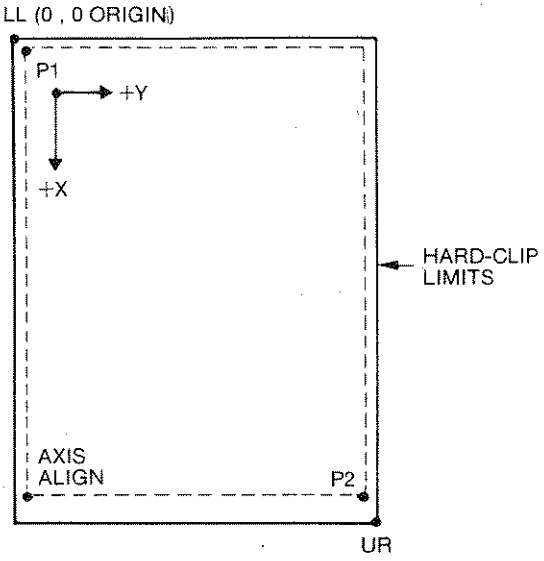


Rotated 90 Degrees
(RO 90 ; or **ROTATE-90**)

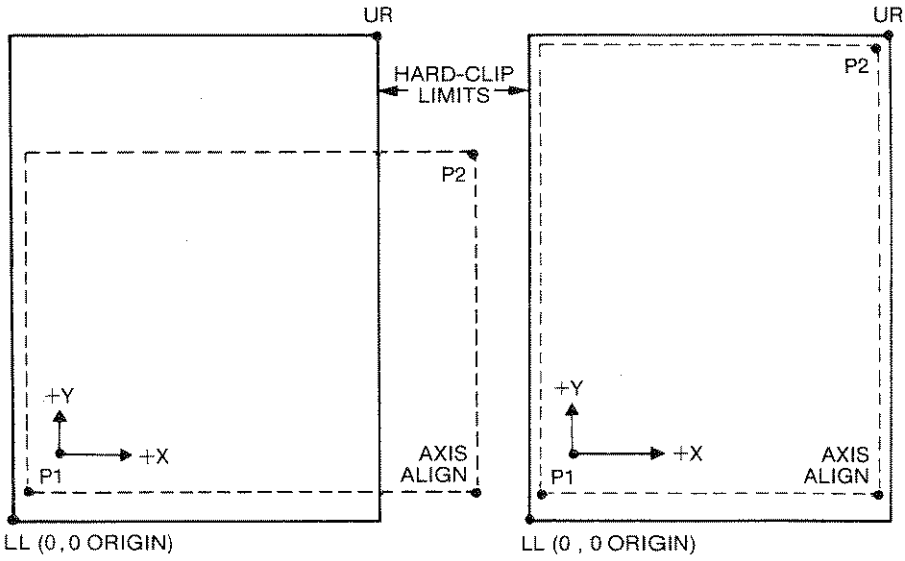
Rotated 90 Degrees
(RO 90 ; IP ; or **ENTER** + **ROTATE-90**)

Forms of Rotation on A4/A-Size Paper

ManualsPlus.com



Default Orientation
(RO ; or RO 0 ; or **ROTATE-0**)



Rotated 90 Degrees
(RO 90 ; or **ROTATE-90**)

Rotated 90 Degrees
(RO 90 ; IP ; or **ENTER** + **ROTATE-90**)

Forms of Rotation on A3/B-Size Paper

ManualsPlus.com

The physical position of the pen does not change when the coordinate system is rotated. Instead, the plotter updates the internal graphics position (X,Y coordinate location) to reflect the new coordinate system. You can obtain the coordinates of the new pen position by executing an OA or OC instruction after the rotation.

The RO instruction remains in effect until the rotation is changed by another RO instruction or from the front panel, or the plotter is initialized. Rotations are not cumulative. In other words, you cannot rotate a plot 90 degrees and then rotate another 90 degrees in order to achieve 180-degree rotation. You can only toggle between 0- and 90-degree rotation.

The following table summarizes the possible error conditions or unexpected results that you might observe with the RO instruction.

Condition	Error	Plotter Response
no parameters	none	establishes default orientation
more than 1 parameter	2	executes first parameter
parameter out-of-range	3	ignores instruction
RO 90; when plot is already rotated 90 degrees	none	ignores instruction

The Input Window Instruction, IW

USES: The IW instruction defines a rectangular area, called the window, that establishes soft-clip limits. Subsequent programmed pen motion will be restricted to this area. Use this instruction when you want to be sure that your data falls within a specific area. For example, you might define a window for a graph on part of the page, and then define a new window on the rest of the page for labels. By establishing windows, you can be certain that stray lines or labels won't appear in the wrong areas.

SYNTAX: *IW* X_1, Y_1, X_2, Y_2 *term*
or
IW *term*

Parameter	Format	Range	Default
X- and Y-coordinates	integer	-2^{23} to $2^{23}-1$	current hard-clip limits (depends on paper size)

ManualsPlus.com

EXPLANATION: The four parameters of the IW instruction specify the X- and Y-coordinates of any two diagonally opposite corners of the window area. The setting of the front-panel **STANDARD** or **ENHANCED** function key determines how the coordinates are interpreted, as follows.

STANDARD — The coordinates are interpreted as plotter units, regardless of whether or not scaling is in effect. Therefore, the window is always an absolute size and does not change when P1 and P2 change.

ENHANCED — If scaling is not active at the time the window is defined, the coordinates are interpreted as plotter units. Therefore, the window is always an absolute size and does not change when P1 and P2 change.

If scaling is active at the time the window is defined, the coordinates are interpreted as user units. When a window is defined in user units, the window will compress or expand in direct proportion to subsequent changes in the scaling points P1 and P2. Turning off scaling after the window is defined does not eliminate or change the window. In this case, clipping will still occur if an attempt is made to plot outside the physical area previously defined in user units. This is because the plotter now monitors that same area by internally converting the same boundaries into plotter units.

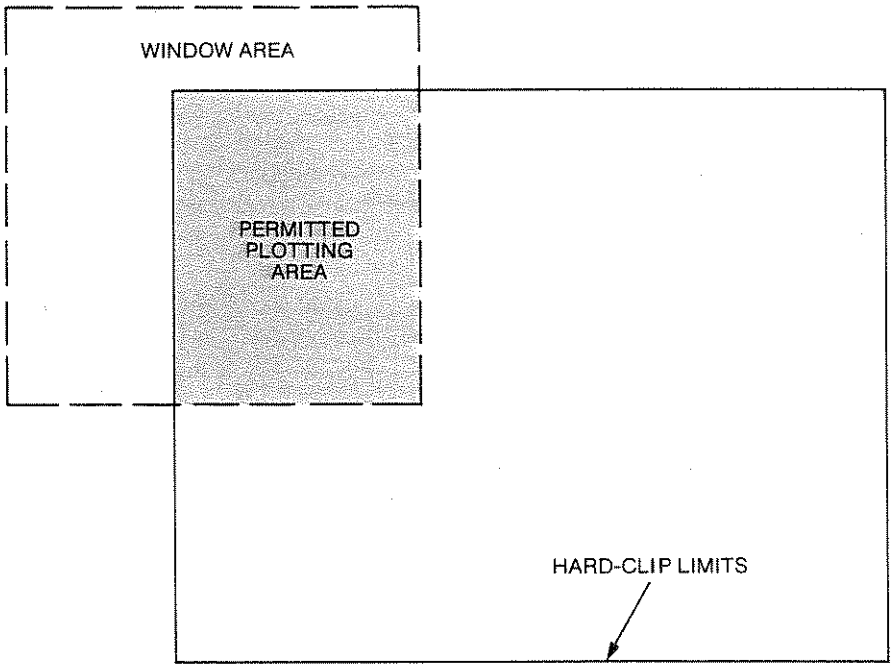
A window can be anywhere inside, outside, or intersecting the hard-clip limits. However, all programmed pen motion is restricted to that part of the window that lies within the hard-clip limits, as shown in the illustration on the next page. For more information, refer to Relationship of Plotting Instructions and Graphics Limits in Chapter 4.

NOTE: Pen motions invoked from the front panel are not restricted by a window. ■

If the window is entirely outside the hard-clip limits, no programmed pen motion can occur. This can happen when:

- The IW instruction defines a window to be outside the hard-clip limits.
- The current window is defined in user units, and a subsequent change in the position of P1 and/or P2 causes the window to move so that it no longer intersects the hard-clip limits.
- A subsequent axes rotation causes the window to fall outside the hard-clip limits.

ManualsPlus.com



An IW instruction without parameters (IW;) defaults the window to be the current hard-clip limits. The window is also set to the hard-clip limits if the window is defined in user units but the coordinates of either corner are out of range when expressed in plotter units. For a discussion of converting user units to plotter units, refer to Appendix A.

The IW instruction remains in effect until another IW instruction is executed, or the plotter is initialized or set to default conditions.

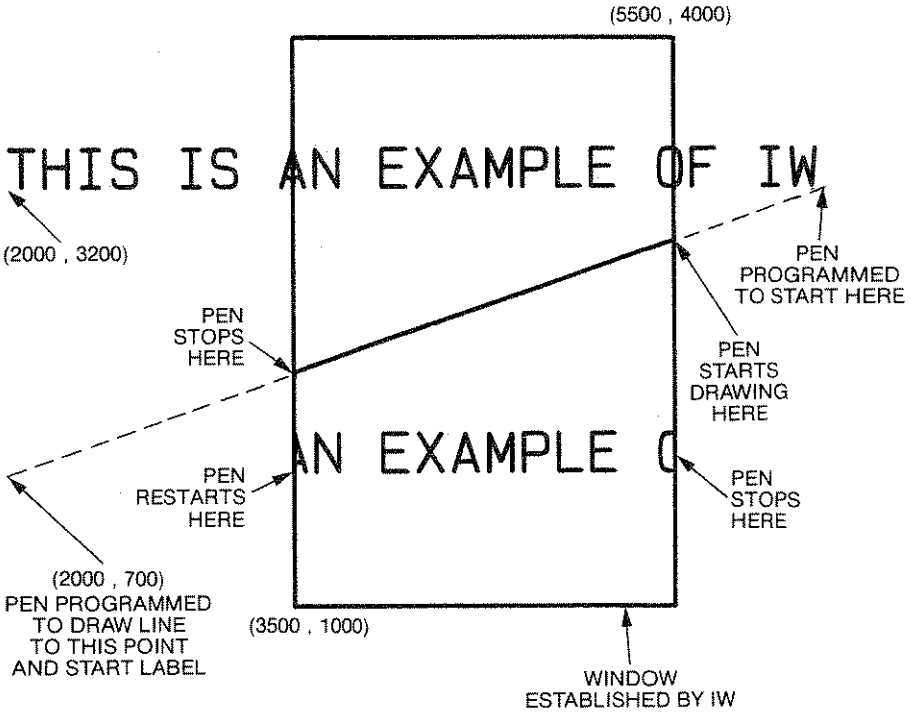
The following table summarizes the possible error conditions or unexpected results that you might observe with the IW instruction.

Condition	Error	Plotter Response
no parameters	none	establishes hard-clip limits as default window
more than 4 parameters	2	executes first 4 parameters
1, 2, or 3 parameters	2	ignores instruction
$X_1 = X_2$ or $Y_1 = Y_2$	3	ignores instruction
parameter out-of-range	3	if scaling is off, ignores instruction; if scaling is on, establishes hard-clip limits for window (see previous discussion in text)

ManualsPlus.com

Example — Effects of Specifying a Window on Labels and Lines

The following example draws a label, then establishes a window and draws the label again along with a line. Notice how the line and label are clipped after the window has been established, but not before. (For an example of using windows to enlarge a portion of a plot, refer to Techniques for Changing the Plotting Area later in this chapter.)




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;"
30  PRINT #1, "SI.3,.5;PA2000,3200;"
40  PRINT #1, "LBTHIS IS AN EXAMPLE OF IW"+CHR$(3)
50  PRINT #1, "IW3500,1000,5500,4000;"
60  PRINT #1, "PD2000,1700;"
70  PRINT #1, "LBTHIS IS AN EXAMPLE OF IW"+CHR$(3)
80  PRINT #1, "PU3500,1000;EA5500,4000;"
90  PRINT #1, "SP0;"
100 END
```

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1

ManualsPlus.com

30 sets the character size; moves to the starting position for the first label

40 draws the label

50 specifies a window 2000 plotter units wide and 3000 plotter units high

60 draws a line from the end of the first label to the beginning of the next label (pen only physically draws within the window)

70 draws the second label (pen only physically draws within the window)

80 moves to the lower-left corner of the window; outlines the window

90 returns the pen to the carousel

The Output Window Instruction, OW

USES: The OW instruction outputs the X,Y coordinates of the lower-left and upper-right corners of the current window in which plotting can occur. Use this instruction when you need to know the size of the current window. This is particularly helpful when the window defined by the IW instruction might be larger than the hard-clip limits, so that the current window is actually limited to the intersection of the hard-clip limits and the window defined by IW.

SYNTAX: *OW term*

RESPONSE: X_{LL}, Y_{LL}, X_{UR}, Y_{UR} [TERM]

EXPLANATION: After an OW instruction is received, the plotter outputs four integers in ASCII, which represent the X,Y coordinates of the lower-left (LL) and upper-right (UR) corners of the current window. If the **STANDARD** function key is on, the coordinates are output as integers in plotter units. If the **ENHANCED** function key is on, the coordinates are output as integers in current units (plotter units if scaling is off or user units if scaling is on). The current window is defined as that portion of the window that intersects with the hard-clip limits (refer to the illustration shown under The Input Window Instruction, IW, presented previously). If no window has been defined with the IW instruction, the plotter outputs the current hard-clip limits.

After you send the OW instruction, your program should read the plotter's output response. Refer to Hints for Obtaining Plotter Output Responses in Chapter 13.

The table on the following page summarizes the possible error conditions or unexpected results that you might observe with the OW instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	outputs current window coordinates
no window currently defined	none	outputs current hard-clip limits
1 or more parameters	2	outputs current window coordinates

The Output Hard-Clip Limits Instruction, OH

USES: The OH instruction outputs the X,Y coordinates of the lower-left and upper-right corners of the current hard-clip limits. Use this instruction to determine the plotter-unit dimensions of the physical area in which plotting can occur. This instruction is particularly helpful to determine the new hard-clip limits when the axes have been rotated and/or when the axis-align function has caused a slight shift in the axes. (Refer to the Operation and Interconnection Manual for information on the axis-align function.)

SYNTAX: *OH term*

RESPONSE: X_{LL}, Y_{LL}, X_{UR}, Y_{UR} [TERM]

EXPLANATION: After an OH instruction is received, the plotter outputs four integers in ASCII, which represent the X,Y coordinates of the lower-left (LL) and upper-right (UR) corners of the current hard-clip limits. The coordinates are always output in plotter units. Leading zeroes are suppressed.

The hard-clip limits determine the maximum physical plotting area, and are initially established when the plotter senses the paper edges while loading paper. Any subsequent IW instruction can further limit the plotting area by establishing soft-clip limits; however, these limits do not affect the hard-clip limits output by the OH instruction.

The default hard-clip limits for each paper size are shown in the following table. If you have used the front-panel axis-align function, the hard-clip limits might be slightly smaller than those shown. The only other operation that will cause different limits to be output is rotating the axes (either from the front panel or by the RO instruction). In this case, the X,Y coordinates are reversed (e.g., 7600 , 10 870 instead of 10 870 , 7600).

ManualsPlus.com

Paper Size	Hard-Clip Limits (Default Axis Orientation)			
	X_{LL}	Y_{LL}	X_{UR}	Y_{UR}
A4	0	0	10870	7600
A3	0	0	15970	10870
A	0	0	10170	7840
B	0	0	16450	10170

After you send the OH instruction, your program should read the plotter's output response. Refer to Hints for Obtaining Plotter Output Responses in Chapter 13.

The following table summarizes the possible error conditions or unexpected results that you might observe with the OH instruction.

Condition	Error	Plotter Response
no parameters	none	outputs current hard-clip limits
1 or more parameters	2	outputs current hard-clip limits

The Output P1 and P2 Instruction, OP

USES: The OP instruction outputs the X,Y coordinates of the current setting of the scaling points P1 and P2. Use this instruction to compute the number of plotter units per user unit when scaling is on, to determine the numeric coordinates of P1 and P2 when they have been set manually from the front panel, or in conjunction with the IW instruction to set the window to P1 and P2.

SYNTAX: *OP term*

RESPONSE: $P1_X$, $P1_Y$, $P2_X$, $P2_Y$ [TERM]

EXPLANATION: After an OP instruction is received, the plotter outputs four integers in ASCII, which represent the X,Y coordinates of P1 and P2 in plotter units. The possible range is -2^{23} to $2^{23}-1$. Upon completion of output, bit position 1 of the status word is cleared (refer to The Output Status Instruction, OS, in Chapter 13).

After you send the OP instruction, your program should read the plotter's output response. Refer to Hints for Obtaining Plotter Output Responses in Chapter 13.

The table on the next page summarizes the possible error conditions or unexpected results that you might observe with the OP instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	outputs current P1 and P2 coordinates
1 or more parameters	2	outputs current P1 and P2 coordinates

Techniques for Changing the Plotting Area

The IP and IW instructions provide a great deal of flexibility for controlling the size of the plotting area and the proportions of a plot. You've already seen this in Chapter 3, where you learned how to set the P1/P2 frame and assign a scale so that isotropic scaling would be established and geometric shapes would be drawn correctly. In this section, you will learn more techniques, such as placing more than one equal-sized plot on a page, enlarging/reducing plots, using windows to magnify portions of plots, and creating mirror images.

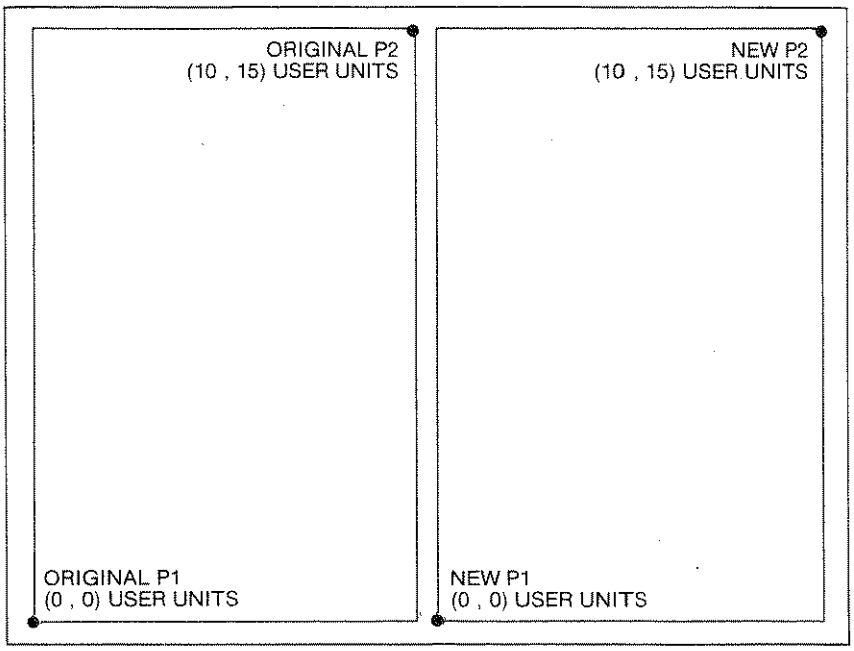
Preparing Equal-Sized Plots on One Page

The easiest way to prepare equal-sized plots on one page is to take advantage of the fact that P2 follows P1 whenever P1 is changed. The following program illustrates the procedure (you could also change P1 and P2 settings from the front panel in the same way).

First, this program specifies a P1/P2 frame for the left-hand side of the page. Then it assigns a scale of 0 to 10 along the X-axis and 0 to 15 along the Y-axis, and draws a rectangle to illustrate the boundaries of the P1/P2 frame. Of course, you could assign any scale and plot anything here. Then the IP instruction specifies a new P1 location; P2 automatically tracks P1, so you don't have to perform the arithmetic to specify a new P2 with the same dimensions as the first P1/P2 frame. Finally, a rectangle is again drawn around the P1/P2 boundaries, using the same scaling as for the first P1/P2 frame.

```
10  'Insert configuration statement here
20  PRINT #1, "IN;IP0,200,4950,7840;"
30  PRINT #1, "SC0,10,0,15;"
40  PRINT #1, "SP1;PA0,0;ER10,15;"
50  PRINT #1, "IP5220,200;"
60  PRINT #1, "PA0,0;ER10,15;"
70  PRINT #1, "SP0;"
80  END
```

ManualsPlus.com



NOTE: These P1/P2 frames are not windows or graphics limits. When drawing the plots for each half of the page, the pen can actually move anywhere within the hard-clip limits. However, since P1 and P2 moved, the scale assigned to them moved with them. This allows you to use the same coordinates on both halves of the page. If you did not assign a scale to P1 and P2 with the SC instruction, you would have to calculate the new plotter-unit coordinates for the plot on the second half of the page. ■

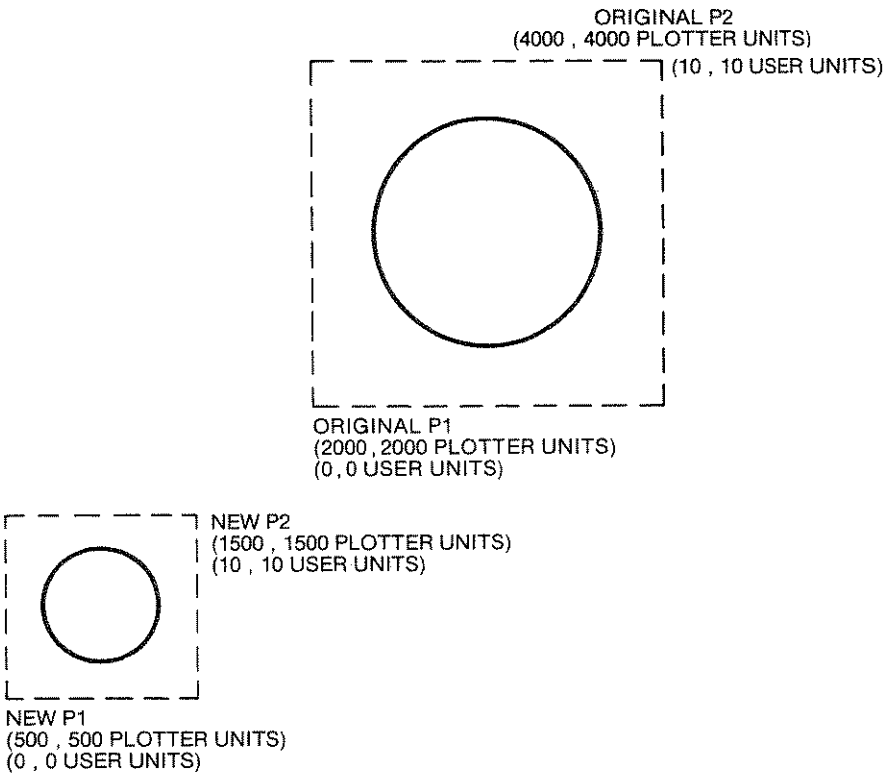
Reducing/Enlarging Plots

The basic technique for changing the size of a plot is to assign a scale to P1 and P2, and then to move P1 and P2 to a smaller or larger portion of the page. As long as scaling is active, the changes in P1 and P2 produce the effect of reducing or enlarging the plot. This is particularly useful when you want to be able to draw your plot on any size of paper, or on any portion of the page (as described above).

The program on the next page is very simple in order to show the concept. First, the program specifies a square P1/P2 frame, and assigns a scale of 0 to 10 along both axes. Then it draws a circle centered in the P1/P2 frame. Next, it specifies a new, smaller P1/P2 frame. Without changing the scaling, it redraws the same circle, again centered. This time the circle is much smaller. To enlarge the circle, specify a new P1/P2 frame that is larger.

ManualsPlus.com

```
10  'Insert configuration statement here
20  PRINT #1, "IN;IP2000,2000,4000,4000;"
30  PRINT #1, "SC0,10,0,10;"
40  PRINT #1, "SP1;PR5,5;CI3;"
50  PRINT #1, "IP500,500,1500,1500;"
60  PRINT #1, "PR5,5;CI3;"
70  PRINT #1, "SP0;"
80  END
```



Using Windows When Scaling Is On to Reduce/Enlarge Portions of a Plot

You can also change the size of a portion of a plot if you use the previous technique in conjunction with the IW instruction. Be sure the **ENHANCED** function key on the front panel is on. When the **ENHANCED** function key is on, the plotter can interpret coordinates as scaled user units. (If the **STANDARD** function key were on, the plotter would interpret coordinates only as plotter units, and changes in P1 and P2 would not affect the size of the window.)

The following example contains a more complex plot so that you can see the windowing and enlarging effects more clearly. Lines 120–290 are a subroutine that draws the axes and plots the function sine π from

ManualsPlus.com

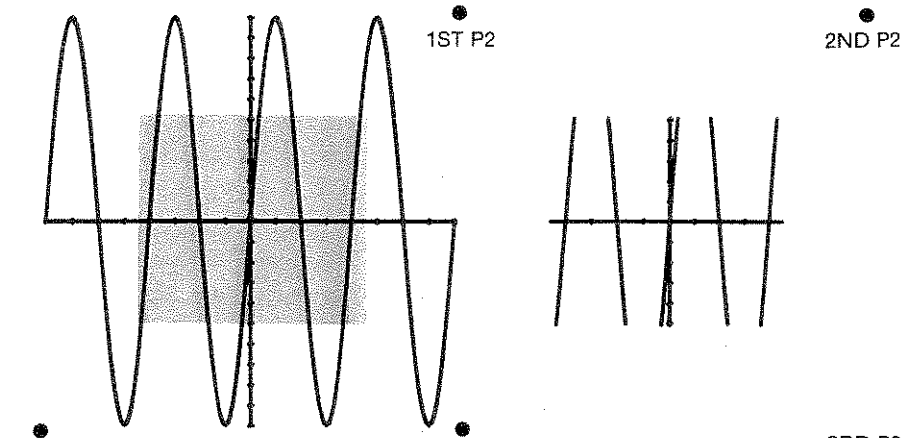
$-4\ \pi$ to $4\ \pi$ (π is approximately equal to 3.14593). This subroutine is called three times:

1. First, the entire plot is drawn without any clipping. However, the plot is drawn only on the upper-left corner of the page because it is drawn in user units that are within the P1/P2 frame established by lines 20 and 30.
2. Second, a new P1/P2 frame is specified in the upper-right corner of the page (line 50). This P1/P2 frame is the same size as the first frame; however, a smaller window is defined in the center of this area (line 60). Now when the subroutine plots the sine function, only that part that lies within the window is plotted. Notice that this portion is the same scale as the first plot.
3. Finally, a larger P1/P2 frame is specified for the bottom center of the page; however, the window is not changed (line 80). Now when the subroutine plots the sine function, the same portion that was plotted in step 2 appears, but it is enlarged.

NOTE: The scaling is always the same; P1 always has the value 0,0 user units and P2 always has the value 6500 , 6500 user units. ■

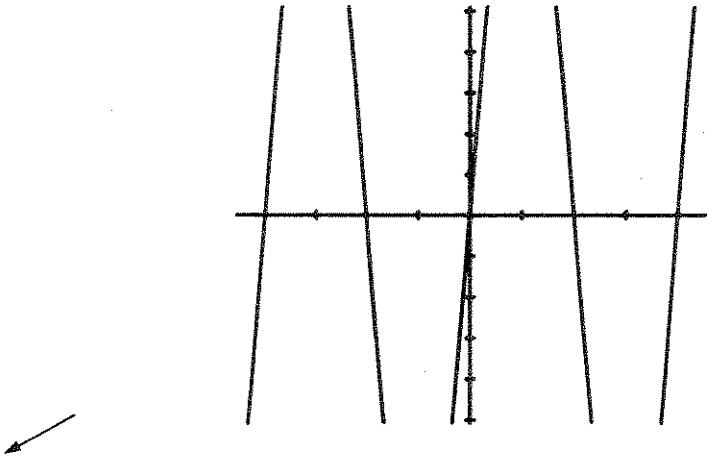
```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;IP1000,5300,3500,7800;"
30  PRINT #1, "SC0,6500,0,6500;"
40  GOSUB 120
50  PRINT #1, "IP3500,5300;"
60  PRINT #1, "IW1425,1625,5075,4875;"
70  GOSUB 120
80  PRINT #1, "IP1100,800,6100,5800;"
90  GOSUB 120
100 PRINT #1, "SP0;"
110 GOTO 300
120 ' Plotting subroutine starts here.
130 PRINT #1, "PA50,3250;PD;XT;"
140 FOR XINTERVAL =1 TO 16
150   PRINT #1, "PR400,0;XT;"
160 NEXT XINTERVAL
170 PRINT #1, "PU;PA3250,50;PD;YT;"
180 FOR YINTERVAL=1 TO 20
190   PRINT #1, "PR0,320;YT;"
200 NEXT YINTERVAL
210 PRINT #1, "PU;"
220 PI=3.141593
230 FOR X=-4*PI TO 4*PI+.1 STEP PI/20
240   XCOORD=6400/25.13*X+3250
250   YCOORD=3200*SIN(X)+3250
260   PRINT #1, "PA";XCOORD;",";YCOORD;";PD;"
270 NEXT X
280 PRINT #1, "PU6500,6500;"
290 RETURN
300 END
```

ManualsPlus.com



Original Call
(Lines 30–50)

Call with Window
(Lines 60–80)



Call with Window Enlarged into New Area
(Lines 90-100)

ManualsPlus.com

Creating Mirror Images

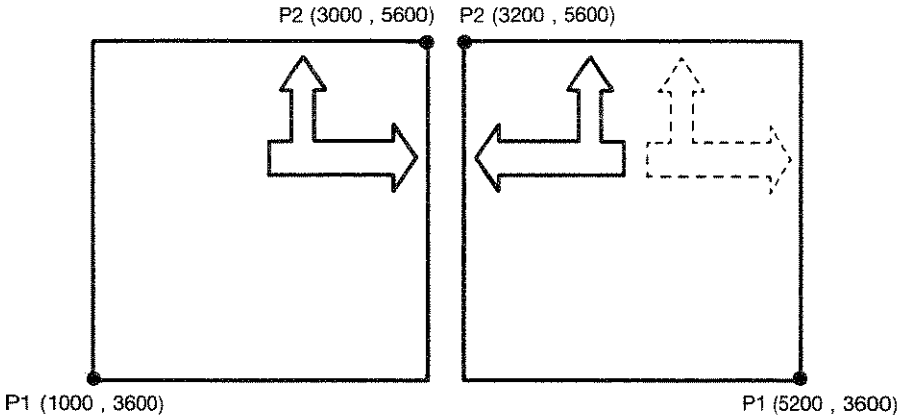
For most applications, you will probably set P1 and P2 so that P1 is in the lower-left corner and P2 is in the upper-right corner of the P1/P2 frame. However, it is possible to change the relationships of P1 to P2. When you do, an interesting phenomenon known as mirror images occurs. Mirror images can happen with all plotting and labeling instructions. Mirror images from labeling instructions can be very complex; these are discussed fully in Parameter Interaction in Labeling Instructions in Chapter 7.

Mirror images from plotting instructions are easier to illustrate. The following program uses a subroutine to be sure that the exact same plot (arrows) is drawn four times. The main program sets P1 and P2, then goes to the plotting subroutine to outline the P1/P2 frame and draw the arrows. It then sets P1 and P2 to be in different corners, and repeats the drawing. This process continues until all four possible combinations of P1 and P2 have been plotted. (The original "normal" plot is shown in each plot so that you can better see the orientation of the mirror image.)

NOTE: The scaling is the same in each case; P1 always has the value −15,−10 user units and P2 always has the value 15,10 user units. ■

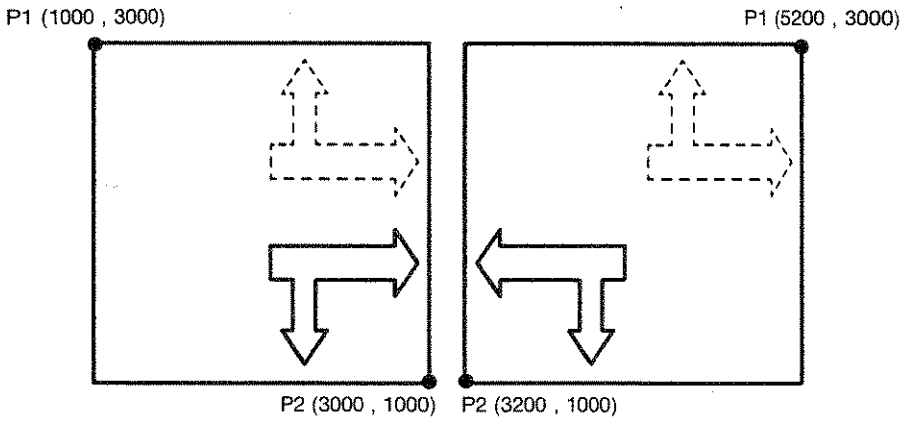
```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;IP1000,3600,3000,5600;"
30  PRINT #1, "SC-15,15,-10,10;"
40  GOSUB 130
50  PRINT #1, "IP5200,3600,3200,5600;"
60  GOSUB 130
70  PRINT #1, "IP1000,3000,3000,1000;"
80  GOSUB 130
90  PRINT #1, "IP5200,3000,3200,1000;"
100 GOSUB 130
110 PRINT #1, "SP0;"
120 GOTO 170
130 PRINT #1, "PU-15,-10;ER15,10;"
140 PRINT #1, "PA1,2;PD1,4,3,4,3,7,2,7,4,9,6,7,5,7;"
150 PRINT #1, "PD5,4,12,4,12,5,14,3,12,1,12,2,1,2;PU;"
160 RETURN
170 END
```

ManualsPlus.com



Normal
(Lines 20–40)

Reversed
(Lines 50–60)



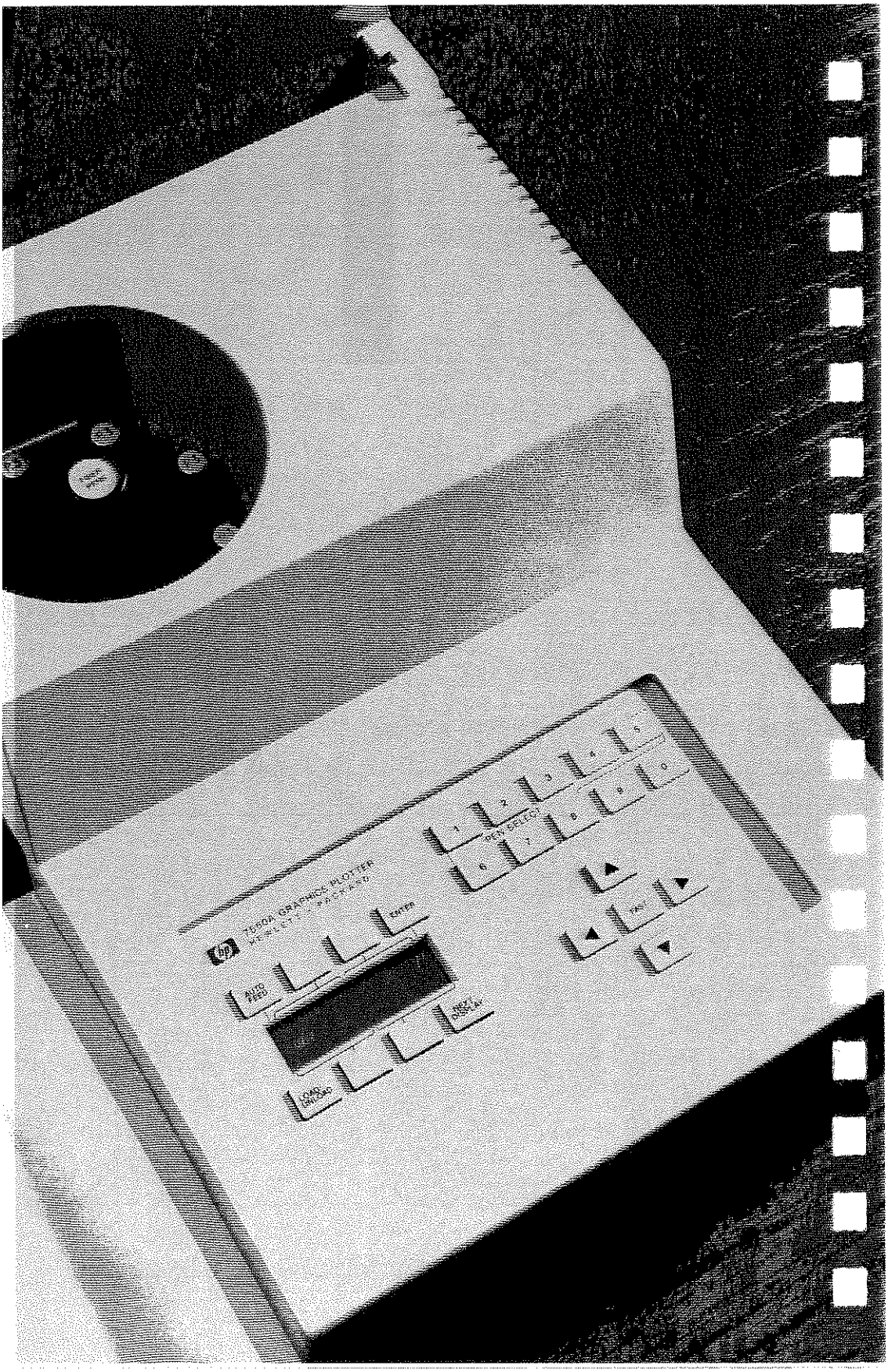
Upside Down
(Lines 70–80)

Upside Down and Reversed
(Lines 90–100)

ManualsPlus.com

Notes

ManualsPlus.com



ManualsPlus.com

Chapter 10
Front-Panel Functions
Pen Control, Paper Feed, Replotting, and Function Keys

What You'll Learn in This Chapter

In this chapter you will learn how to control many of the plotter's front-panel functions using HP-GL instructions in your programs. For example, you can change the pen force, acceleration, and velocity. You can also cause the paper to load automatically and replot your drawings up to 99 times automatically. Finally, you can define the plotter's function keys to print messages on the front-panel display.

HP-GL Instructions Covered

AP The Automatic Pen Operations Instruction
GP The Group Pen Instruction
SG The Select Pen Group Instruction
FS The Force Select Instruction
AS The Acceleration Select Instruction
VS The Velocity Select Instruction
CV The Curved Line Generator Instruction
PG The Page Feed Instruction
NR The Not-Ready Instruction
BF The Buffer Plot Instruction
RP The Replot Instruction
WD The Write to Display Instruction
KY The Define Key Instruction
OK The Output Key Instruction
GC The Group Count Instruction
OG The Output Group Count Instruction

ManualsPlus.com

Terms You Should Understand

Force — the pressure with which the pen touches the plotting surface (measured in grams).

Acceleration — the rate at which the pen attains its maximum velocity (measured in g's, which are a measure of metres per second per second).

Velocity (Speed) — the rate at which the pen moves (measured in centimetres per second).

Poll — to request information from the plotter. For example, you might poll the plotter to discover whether a sheet of paper was loaded successfully before starting the plot.

Interactive Programming — a technique that allows two-way communication between an operator and a device such as the plotter or a computer. Interactive programs often communicate by displaying a message that requires operator action (such a pressing a function key) before the program will continue. The write to display and define key instructions, WD and KY, provide a method for you to design interactive programs for the plotter.

Keyboard Mode — a mode established by the write to display instruction, WD. When the plotter is in keyboard mode, the front-panel display and the four unlabeled function keys are available for use in a program. This means that a program can cause a message to be displayed, assign a specific function to one of the keys, or detect when one of the keys has been pressed.

Spooling — a technique by which data destined for the plotter is placed in a queue on a mass storage device to await transmission. The group count instructions, GC and OG, aid in allowing data to be spooled to the plotter.

The Automatic Pen Operations Instruction, AP

USES: The AP instruction controls certain automatic pen operations, such as the automatic storage feature that returns a pen to the carousel if the pen has been in the pen holder without drawing for an allotted time. Use this instruction to turn off automatic pen operations when you are using the digitizing sight, or to reestablish pen operations when you resume plotting.

SYNTAX: *AP* n *term*
or
AP *term*

ManualsPlus.com

Parameter	Format	Range	Default
n	integer	0–143	7

EXPLANATION: The parameter n consists of the sum of the values of the automatic pen operations you want to turn on.

0 disables automatic pen operations. Pens not lifted until commanded by a Pen Up instruction or the front panel. Pens not stored until commanded by an SP0 instruction or the front panel. Pens retrieved immediately when selected by an SP instruction.

1 lifts the pen if it has been down for the allotted time.*

2 stores the pen if it has not moved in the allotted time.* If unable to put the pen away, lifts the pen (if down).

4 does not retrieve a pen selected by the SP instruction until the new pen is required to draw.

8 merges all pen up moves.

128 automatically converts all SP instructions to SG instructions so you can group pens with software that does not use the SG instruction.

To turn on any subset of the pen operations, add the value of the operations. If any values are not included in the sum, the associated operations are turned off. For example, AP 3; turns on only operations 1 and 2. AP 7; turns on operations 1, 2, and 4; this function is the same as entering AP; and is the default operation set for the plotter. AP 143; turns on all the automatic pen operations.

An AP instruction remains in effect until a new AP instruction is received or the plotter is initialized or set to default conditions.

NOTE: Any pen select operation invoked from the front panel will be executed immediately. Also, if you insert a pen directly into the pen holder, the plotter cannot detect it until a pen select operation causes it to interact with the carousel and update its status. Therefore, none of the automatic pen operations will function until the pen is detected. ■

*The "allotted time" depends on the carousel in the plotter. If no carousel is in the plotter or if a paper or roller-ball carousel is loaded, the allotted time is 65 seconds. If a transparency or drafting carousel is loaded, the allotted time is 15 seconds.

ManualsPlus.com

The following table summarizes the possible error conditions or unexpected results that you might observe with the AP instruction.

Condition	Error	Plotter Response
no parameters	none	turns on 1, 2, and 4
more than 1 parameter	2	executes first parameter
parameter does not represent a valid sum	3	ignores instruction
parameter out-of-range	3	ignores instruction

The Group Pen Instruction, GP†

USES: The GP instruction specifies which pens belong to a designated group. Use this instruction with pens of the same color and width to extend the effective writing distance beyond the limits of one pen.

SYNTAX: *GP* (group number (,pen number (,number of pens (,length)))) *term*

PARAMETERS:

Parameter	Format	Range	Default
group number	integer	1–8	all groups
pen number	integer	1–8	specified group number
number of pens	integer	1–8*	1
length	integer	1–50 000	100

*pen number + number of pens must be < 9

EXPLANATION: The parameters used in the GP instruction are explained below.

1. Group number. The first parameter designates the group to be defined and is used in conjunction with the SG instruction.
2. Pen number. The second parameter is the number of the *first* pen in the group. When a pen number is omitted, a group will consist of one pen whose number is the same as the group number.

†This instruction applies to plotters whose serial number prefix is 2629A or larger.

ManualsPlus.com

3. Number of pens. The third parameter is the total number of pens in the group. The group begins with the pen specified in parameter two and increases sequentially by one until the total number equals the third parameter.

   The pen sequence will stop with pen 8, even if the number of pens in the sequence is less than the number specified in parameter three. (See the examples below.)

   If the number of pens in a group is omitted, the group will default to one pen.

4. Length. The fourth parameter is the length in metres at which the pens will switch. Numbers larger than 50 000 will be clipped to 50 000. When omitted, the parameter defaults to 100.

At power-up, or when the plotter is initialized (IN ;), or when the GP instruction is issued without parameters (GP ;), *all* groups will consist only of the corresponding pen. For example, GP ; is equivalent to GP 1 , 1 ; GP 2 , 2 ; GP 3 , 3 ; etc.

Specifying one parameter causes *only* the group specified to point to one pen. For example, GP 1 ; is equivalent to GP 1 , 1 , 1 ;.

Specifying two parameters maps a group to the pen specified. For example GP 2 , 3 ; means that group 2 consists only of pen 3.

Below are some additional examples.

GP 2 , 7 , 4 ;	Group 2 will contain pens 7 and 8 and will switch every 100 metres (default).
GP 9 , 2 , 3 ;	NOP'd.
GP 1 , 9 , 3 ;	NOP'd. An SG 1 ; instruction would also be NOP'd.
GP 2 , 2 , 2 , 5000 ;	Group 2 will contain pens 2 and 3 and and will switch every 5000 metres.
GP 1 , 1 , 1 , 300 ;	Group 1 will contain only pen 1, so the switch distance is irrelevant.

None of the above examples will cause any errors. However, if any of the parameters of the GP instruction are less than one, the instruction is ignored and the group being defined will not change.

ManualsPlus.com

The Select Pen Group Instruction, SG†

USES: The SG instruction enables the plotter to alternate between pre-designated pens within a group. Use this instruction with the GP instruction to extend the effective writing distance beyond the limits of one pen.

SYNTAX: *SG* group number *term*

PARAMETERS:

Parameter	Format	Range	Default
group number	integer	0 to 8	0 (no pen)

EXPLANATION: The parameter applies to a pen group previously designated with the GP instruction. The SG instruction selects the pen or pens which belong to this corresponding group. When the group contains more than one pen, the plotter will sequentially select pens according to the length parameter of the GP instruction.

When no groups have been previously defined with a GP instruction, the SG instruction is equivalent to a Select Pen Instruction, SP. Accordingly, SG 0 ; and SG ; are equivalent to SP 0 ;: any pen in the pen holder will be placed in the carousel.

If an out-of-range parameter is used, the instruction will be ignored. A parameter less than 0 will also generate error 3.

The Force Select Instruction, FS

USES: The FS instruction establishes the force with which the pen draws on the plotting surface. Use this instruction to set individual forces for pens when mixing more than one type of pen in the same carousel. This optimizes pen life and line quality for each pen.

SYNTAX: *FS* pen force (, pen number) *term*
or
FS *term*

Parameter	Format	Range	Default
pen force	integer	1–8	depends on carousel type
pen number	integer	1–8	all pens

EXPLANATION: The following paragraphs describe details of each parameter.

†This instruction applies to plotters whose serial number prefix is 2629A or larger.

ManualsPlus.com

1. Pen Force. The pen force parameter is an integer that represents specific forces in grams, as follows.

1 = 15 grams	5 = 45 grams
2 = 24 grams	6 = 51 grams
3 = 30 grams	7 = 57 grams
4 = 36 grams	8 = 66 grams

2. Pen Number. The pen number parameter represents the pen that will be assigned the specified pen force. If no pen number is given, the force will be assigned to all pens in the carousel.

An FS instruction without parameters (FS;) establishes the default force value for the carousel that is currently installed in the plotter, as follows.

Carousel Type	Pen Force Parameter	Pen Force in Grams
Drafting pen	1	15
Paper and transparency fiber-tip pens	2	24
Roller-ball pen	6	51

The FS instruction remains in effect until a new FS instruction is executed, a new force is selected from the front panel, or the plotter is initialized. Also, if a new carousel with a different default force is installed, all pens default to the new force.

The following table summarizes the possible error conditions or unexpected results that you might observe with the FS instruction.

Condition	Error	Plotter Response
no parameters	none	establishes default force for all pens in current carousel
pen force > 8	none	establishes force 8 (66 grams)
more than 2 parameters	2	executes first 2 parameters
any parameter ≤ 0 or pen number > 8	3	ignores instruction

The Acceleration Select Instruction, AS

USES: The AS instruction establishes the acceleration at which the pen moves. The default 6-g acceleration is suitable for all recommended

ManualsPlus.com

pen and media combinations. Slowing down the acceleration could improve line quality if you are using media that are heavier than recommended.

SYNTAX: *AS* pen acceleration (, pen number) *term*
or
AS *term*

Parameter	Format	Range	Default
pen acceleration	integer	1-6	6
pen number	integer	1-8	all pens

EXPLANATION: The following paragraphs describe details of each parameter.

1. Pen Acceleration. The pen acceleration parameter is an integer that represents the actual acceleration in g's.
2. Pen Number. The pen number parameter represents the pen that will be assigned the specified pen acceleration. If no pen number is given, the acceleration will be assigned to all pens in the carousel.

An AS instruction without parameters (AS ;) establishes the default acceleration value of 6 g's for all carousels. The AS instruction remains in effect until a new AS instruction is executed or the plotter is initialized. Also, if a new carousel is installed, all pens default to 6 g's.

The following table summarizes the possible error conditions or unexpected results that you might observe with the AS instruction.

Condition	Error	Plotter Response
no parameters	none	establishes 6 g's acceleration for all pens
pen acceleration > 6	none	establishes 6 g's acceleration
more than 2 parameters	2	executes first 2 parameters
any parameter ≤ 0 or pen number > 8	3	ignores instruction

The Velocity Select Instruction, VS

USES: The VS instruction establishes the speed (velocity) at which the pen moves. Use this instruction to set individual speeds for pens when mixing more than one type of pen in the same carousel. This can optimize pen life and line quality for each pen. Also, slowing down the pen speed can produce slightly denser lines on any plotting medium.

ManualsPlus.com

SYNTAX: *VS* pen speed (, pen number) *term*
or
VS *term*

Parameter	Format	Range	Default
pen speed	integer	1-80	depends on carousel type
pen number	integer	1-8	all pens

EXPLANATION: The following paragraphs describe details of each parameter.

1. Pen Speed. The pen speed parameter is an integer that represents the actual speed in cm/s. The selected pen speed only applies when the pen is down. Pen movement with the pen up is always performed at 80 cm/s.
2. Pen Number. The pen number parameter represents the pen that will be assigned the specified pen speed. If no pen number is given, the speed will be assigned to all pens in the carousel.

A VS instruction without parameters (VS;) establishes the default speed value for the carousel that is currently installed in the plotter, as follows.

Carousel Type	Pen Speed (cm/s)
Transparency fiber-tip pen	10
Drafting pen	15
Paper fiber-tip pen	50
Roller-ball pen	60

The VS instruction remains in effect until a new VS instruction is executed, a new speed is selected from the front panel, or the plotter is initialized. Also, if a new carousel with a different default speed is installed, all pens default to the new speed.

The table on the next page summarizes the possible error conditions or unexpected results that you might observe with the VS instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	establishes default speed for all pens in current carousel
pen speed > 80	none	establishes 80 cm/s speed
more than 2 parameters	2	executes first 2 parameters
any parameter ⩽ 0 or pen number > 8	3	ignores instruction

The Curved Line Generator Instruction, CV

USES: The CV instruction collects vectors (line segments) in the vector buffer so that they can be plotted as a group. This causes the pen to plot in a continuous motion, rather than stopping and starting at each vector endpoint. As a result, curves appear smoother. Use this instruction to increase the line quality of curves. For example, you could use CV for curves that are approximated by many short line segments, such as arcs, circles, and characters. Or, you could use it when your program must perform computations for each vector, which would ordinarily cause the pen to pause between each vector.

SYNTAX: *CV* n (, input delay) *term*
or
CV *term*

Parameter	Format	Range	Default
n	integer	0 or 1	0
input delay	integer	-2^{23} to $2^{23}-1$ milliseconds	100 ms

EXPLANATION: A CV instruction without parameters (CV;) turns the curved line generator off. This is the same as CV 0;. The following paragraphs describe each parameter.

1. n. This parameter is an integer that controls the curved line generator as follows:

   n = 0 Turns off the curved line generator.
   n = 1 Turns on the curved line generator.

2. Input Delay. The input delay is the amount of time that can elapse from the last vector received before the curved line generator will plot the collected vectors. This parameter is optional, and is ignored

ManualsPlus.com

when the first parameter is 0 (i.e., when the curved line generator has been turned off).

The input delay should be large enough to accommodate the longest time that will elapse between vectors being received by the vector buffer. The default of 100 milliseconds should be sufficient for most computer systems. However, the length of the input delay depends on a number of factors, such as the time required for the controller to evaluate and send data to the plotter, and the time required for the plotter to evaluate data. If necessary, increase the input delay.

If you specify a negative input delay, the plotter sets the delay to 0 milliseconds.

When the curved line generator is in effect, the plotter gathers vectors in the vector buffer until one of the following conditions is met:

- A pen up or pen down instruction, PU or PD, is received.
- The vector buffer becomes full.
- A vector has not been received in the amount of time established by the input delay.
- An output actual position and pen status instruction, OA, is received.

When one of the preceding conditions is met, the vectors that are currently in the vector buffer are released to be plotted. The vectors are then plotted sequentially without stopping. Note that pen acceleration is reduced to 3.5 g's, and pen speed is automatically adjusted according to the sharpness of the turning angle from one vector to the next. Therefore, overall plot throughput will decrease (the plot will be completed less quickly).

Before executing the CV instruction, be sure to allocate memory in the vector buffer using the GM or ESC.T instruction. If the buffer is not large enough, the effects of using the CV instruction are negligible. The default vector buffer size is 44 bytes, which is enough for four vectors when the curved line generator is on. (Each vector requires 10 bytes of buffer space.) To optimize the performance of the curved line generator, estimate the largest number of vectors that a curve is likely to have. For example, 72 vectors would be required to draw a 360-degree circle with 5-degree chords. In this case, you would increase the vector buffer to at least 720 bytes (72 vectors $\times$ 10 bytes per vector). (Refer to The Graphics Memory Instruction, GM, in Chapter 3 or The Allocate Configurable Memory Instruction, ESC.T, in Chapter 14 for details.)

If plot throughput is important, do not increase the vector buffer and input delay too much, because this could cause the plotter to spend unnecessary time waiting for vectors before plotting them. Also, a large

ManualsPlus.com

vector buffer can cause unusually long delays before the plotter responds after you have pressed a front-panel function key (such as **P1**, **P2**, **SPEED**, or **FORCE**). This is because most front-panel functions are added to the vector buffer, which means they will be executed in sequence *after* any vectors currently in the buffer.

The CV instruction remains in effect until a new CV instruction is executed, or the plotter is initialized.

The following table summarizes the possible error conditions or unexpected results that you might observe with the CV instruction.

Condition	Error	Plotter Response
no parameters	none	turns off curved line generator
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction

The Page Feed Instruction, PG (or AF or AH)

USES: The PG instruction unloads the current paper and then automatically loads a new sheet of paper. Use this instruction to cause paper to feed automatically for unattended plotting.

SYNTAX: *PG* n *term*
or
PG *term*
or
AF *term*
or
AH *term*

Parameter	Format	Range	Default
n	integer	-2^{23} to $2^{23}-1$	none

EXPLANATION: The three forms of the page feed instruction are interpreted as follows.

1. PG n;. Any integer included as a parameter in the PG instruction forces a page feed regardless of whether or not the paper has been plotted on. You can use any integer within the plotter's range.

   Executing a PG n; instruction is equivalent to pressing **LOAD** on the front panel when the plotter is in automatic feed mode.

ManualsPlus.com

2. PG;. A PG instruction without parameters causes a new page to load only if the current page has been plotted on.
3. AF; or AH;. An AF or AH instruction without parameters causes a new page to load only if the current page has been plotted on. The AF and AH instructions are included for compatibility with other HP plotters.

In order for any page feed instruction to work, paper must be currently loaded, the plotter must be in the automatic feed mode (an * must appear in the display below the **AUTO FEED** key), and a media loading tray (filled with paper) must be installed. Depending on the form of the PG (or AF or AH) instruction that you use, if these conditions are met, the current page unloads to the rear and the new page is loaded from the media loading tray.

If no paper is loaded when the plotter receives the page feed instruction, it ignores the instruction and enters the not-ready state. If the plotter is in manual feed mode when it receives a page feed instruction, it unloads any paper currently loaded, and enters the not-ready state. At this point, no plotting can occur until paper is loaded, which causes the plotter to exit the not-ready state.

If the media loading tray is empty when the plotter receives a page feed instruction, the message **PAPER LOAD FAILED** appears on the front-panel display and the plotter enters the not-ready state. Simply load the tray with paper, and execute the page feed instruction again.

You can use the ESC.O instruction to poll the plotter to determine whether the plotter has entered the not-ready state.

The table on the next page summarizes the possible error conditions or unexpected results that you might observe with the PG (or AF or AH) instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters (PG, AF, or AH), or 1 or more parameters (AF or AH)	none	forces automatic page feed only if current paper has been plotted on
1 or more parameters (PG only)	none	forces automatic page feed
parameter out-of-range (PG only)	3	ignores instruction
plotter not in automatic feed mode	none	unloads paper and enters not-ready state
media loading tray empty	none	displays **PAPER LOAD FAILED**, unloads paper, and enters not-ready state
no paper currently loaded	none	ignores instruction and enters not-ready state

The Not-Ready Instruction, NR

USES: The NR instruction unloads paper and places the plotter in the not-ready state. Use this instruction at the end of your plotting program to ensure that the next program does not start to draw over your plot.

SYNTAX: *NR term*

EXPLANATION: Executing NR; is the same as pressing **UNLOAD** on the front panel. If a sheet of plotting medium is loaded when the NR; instruction is executed, the plotter unloads the medium as follows: in manual feed mode, the medium is moved toward the front of the plotter before being unloaded; in automatic feed mode, the medium is ejected to the rear of the plotter while being unloaded. The plotter then enters the not-ready state. If no plotting medium is loaded, the plotter is already in the not-ready state.

To exit the not-ready state and allow subsequent plotting, load a sheet of plotting medium in the plotter.

The following table summarizes the possible error conditions or unexpected results that you might observe with the NR instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	unloads paper and enters not-ready state
1 or more parameters	2	unloads paper and enters not-ready state

The Buffer Plot Instruction, BF

USES: The BF instruction causes the plotter to store subsequent plotting instructions while they are being plotted. Use this instruction before the replot instruction, RP, to cause multiple plots to be made automatically.

SYNTAX: *BF term*

EXPLANATION: The BF instruction causes all subsequent plotting instructions to be stored in the replot buffer until an RP instruction is executed. The plot is drawn once while it is being stored. Only plotting instructions are stored; all output and device-control instructions are executed normally during the storage process.

Before executing the BF instruction, you should allocate memory to the replot buffer using the GM or ESC.T instruction. To do so, determine the total number of bytes that your program requires for the other buffers (e.g., the polygon and downloadable character buffers). Then subtract this total number from 12800, and allocate the remainder to the replot buffer. This ensures that the maximum possible memory is available for replotting. The default size of the replot buffer is 9954 bytes. Refer to The Graphics Memory Instruction, GM, in Chapter 3 or The Allocate Configurable Memory Instruction, ESC.T, in Chapter 14.

The BF instruction remains in effect until an RP instruction is executed, the replot buffer overflows, or the plotter is initialized. If the replot buffer overflows before an RP instruction is received, the plotter finishes drawing the plot, but empties the replot buffer and exits the buffer plot mode.

The table on the next page summarizes the possible error conditions or unexpected results that you might observe with the BF instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	stores plotting instructions in replot buffer while drawing them
1 or more parameters	2	stores plotting instructions in replot buffer while drawing them
replot buffer overflow	7	clears the replot buffer and exits buffer plot mode, but completes the first plot

The Replot Instruction, RP

USES: The RP instruction causes the plotter to replot the drawing that is currently stored in the replot buffer. The drawing can be replotted up to 99 times. Use this instruction after the Buffer Plot Instruction, BF, to draw multiple plots automatically.

SYNTAX: *RP* n *term*

Parameter	Format	Range	Default
n	integer	1–99	1

EXPLANATION: The RP instruction causes the currently buffered plot to be redrawn the number of times specified by the parameter n. If the replot buffer is empty, the plotter ignores the RP instruction (refer to the BF instruction, described previously). If there is a plot in the buffer, the response to the instruction depends on whether the plotter is in manual or automatic feed mode, as follows.

- Manual feed mode — At the end of the plot, the plotter unloads the paper to the front. To obtain another plot, you must load a new sheet of paper and execute a new RP instruction or press **REPLOT** and **START** on the front panel. For large quantities, it is more efficient to be sure the plotter is in automatic feed mode.
- Automatic feed mode — At the end of each plot, the plotter unloads the plot to the rear and feeds a new sheet from the media loading tray. It also displays a message on the front panel that indicates how many plots have been completed and how many plots were requested.

  If the media loading tray is empty, the plotter enters the not-ready state. To obtain more plots, you must place more paper in the tray and execute a new RP instruction or press **REPLOT** and **START** on the front panel. You can use the ESC.O instruction in your program to sense when the plotter runs out of paper.

ManualsPlus.com

The following table summarizes the possible error conditions or unexpected results that you might observe with the RP instruction.

Condition	Error	Plotter Response
no parameters	none	plots contents of replot buffer 1 time
manual feed mode	none	plots contents of replot buffer 1 time
media loading tray empty	none	displays **PAPER LOAD FAILED** and enters not-ready state
replot buffer empty	none	ignores instruction
parameter out-of-range	3	ignores instruction

The Interactive Front-Panel Display

You can design interactive plotting programs by sending messages to the front-panel display. For example, you might want the plotter operator to change pens at a certain point in the program. Your program could pause and send the message "CHANGE PENS AND PRESS A KEY" to the display. Then, when a key has been pressed, your program could continue. The plotter has several instructions that allow you flexibility in designing interactive messages for your programs. The instructions are listed next, and are presented in more detail following this section.

WD The Write to Display Instruction
KY The Define Key Instruction
OK The Output Key Instruction
GC The Group Count Instruction
OG The Output Group Count Instruction

There are primarily two methods that you might use when designing interactive programs. One is described in the example above. It involves sending a message that requires a function key to be pressed, then polling the plotter to determine when a key has been pressed. When the response is positive, you might have a subroutine in your program to perform an operation related to the message. To implement this method, you would use the WD instruction to send the message to the display, then use the OK instruction to poll the plotter for a pressed key. This method is illustrated later, under the description of the OK instruction.

The other method involves sending a message to the front-panel display and redefining one or more of the function keys to perform any of 12 specific functions. (The function keys are the four unlabeled keys in the

ManualsPlus.com

center of the front-panel display.) A typical example is to define a function key to perform an "escape" when it is pressed. This would allow an operator to escape from the program, for example, if he discovered that he had loaded the wrong paper or pens; then he wouldn't have to wait for the whole plot to finish before he could correct his mistake. The program would poll the plotter for the escape function being activated. When the program detected the escape function, it would begin "flush mode," which causes all subsequent HP-GL instructions to be discarded until flush mode is terminated. Then the program could automatically start over, or it could terminate until the operator started it again.

To do this type of interactive program, you would use the KY instruction to assign the new function (such as "escape") to one of the four function keys. Then you would use the WD instruction to send a message to the display (such as the word "ESCAPE" to label which key will perform the escape function). Finally, you would send the OG instruction to determine whether the operator has activated the escape function, and to enable flush mode if he has. This method is illustrated later, under the description of the KY instruction.

The Write to Display Instruction, WD

USES: The WD instruction causes the specified message to be displayed on the front panel of the plotter and establishes keyboard mode. Use this instruction when designing interactive programs.

SYNTAX: *WD* c ... c *term*
or
WD *term*
(where *term* is the label terminator defined by the DT instruction)

Parameter	Format	Range	Default
c ... c (up to 32 characters are displayed)	label	any character from decimal code 32 to 95	none

EXPLANATION: When you send a WD instruction with characters, the plotter displays the specified characters and establishes keyboard mode (described below). The plotter displays up to 32 characters (2 lines of 16 characters each); any excess characters are ignored. The characters are displayed using LCD (liquid-crystal display) ANSI ASCII English characters. Therefore, the current character set has no effect on the WD instruction. You can send the actual character or use a function such as CHR$ to specify the decimal code of a character. Only characters with

ManualsPlus.com

decimal codes of 32 to 95 can be displayed; these include many keyboard symbols, numerals, and uppercase letters. (Refer to Appendix A for ASCII character codes.)

NOTE: The WD instruction is similar to other labeling instructions in that it requires a label terminator. The default label terminator is **ETX** (decimal code 3), but may be redefined by the DT instruction. Refer to The Label Instruction, LB, and The Define Terminator Instruction, DT, in Chapter 7 for examples of using label terminators. ■

While in keyboard mode, the message is displayed and the function keys are available for use in a program. This means that you can use the OK or ESC.O instructions to detect whether a function key has been pressed, or you can use the KY instruction to redefine the function keys.

When a function key is pressed while the plotter is in keyboard mode, bit 9 of the extended status word is set. You can detect which key has been pressed with the OK instruction. Or, you can use the ESC.O device-control instruction to detect only that a function key has been pressed (but not which one). Thus, you can set up your program to poll for a key being pressed before continuing. Refer to The Output Key Instruction, OK, for an example.

Alternatively, you can redefine the function keys using the KY instruction. In this case, the plotter will perform the redefined function when the associated function key is pressed. Refer to The Define Key Instruction, KY, for an example.

If an error occurs while the plotter is in keyboard mode, the error overwrites the WD message and temporarily disables keyboard mode. To return to the keyboard mode, press any front-panel function key, or clear the error by sending an OE instruction.

To temporarily exit keyboard mode, press **ENTER** followed by **NEXT DISPLAY**. You can return to the WD display and keyboard mode by pressing **SOFTKEY**. To permanently turn off keyboard mode, send a WD instruction without characters.

The WD instruction remains in effect until another WD instruction overwrites the message, a WD instruction without characters clears the display, or the plotter is initialized. The DF instruction does not affect the WD instruction.

Note that a new WD instruction overwrites the previous one. If the new WD instruction is shorter, you must end it with enough spaces to completely overwrite the first one.

The table on the next page summarizes the possible error conditions or unexpected results that you might observe with the WD instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no characters	none	disables keyboard mode; front panel operates normally
more than 32 characters	none	displays first 32 characters
invalid character	none	if decimal code 0–31, ignores the character; if decimal code 96–126, subtracts 32 and displays that character (e.g., A for a)
error message on display when plotter receives WD instruction	none	displays error until function key is pressed or OE clears the error; then displays WD message
error occurs while in keyboard mode	none	temporarily disables keyboard mode until function key is pressed or OE clears the error; then returns to keyboard mode

The Define Key Instruction, KY

USES: The KY instruction assigns a predefined function to one of the front-panel function keys. Use this instruction in conjunction with the WD instruction when designing interactive programs.

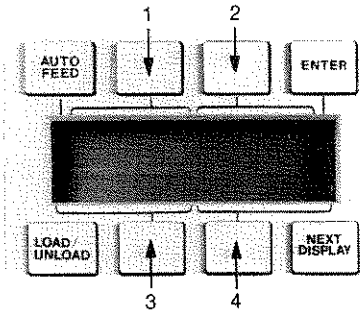
SYNTAX: *KY* key (, function) *term*
or
KY *term*

Parameter	Format	Range	Default
key	integer	1–4	none
function	integer	0–12	none

EXPLANATION: The parameters are interpreted as follows.

1. Key. The key parameter specifies which of the four function keys will take on a new definition. The function keys are numbered as shown in the following illustration.

ManualsPlus.com



2. Function. The function parameter defines which function will be assigned to the specified key. You can choose among 12 functions, as listed below. The specified function operates in the same manner as it does when in normal operating mode. For example, if you specify the "pen up" function for key 4 (KY 4,2;), then the plotter will lift the pen each time key 4 is pressed, just as it does when you press the normal **PEN-UP** key.

0	cancel key	7	rotate
1	view	8	force
2	pen up	9	align
3	pen down	10	reset
4	P1	11	clear
5	P2	12	escape
6	speed		

To cancel one key, specify the key parameter and then omit the function parameter, or specify 0 (zero) for the function parameter. To cancel all keys, send the KY instruction without parameters (KY;). When you cancel a key, it returns to an undefined state.

The KY instruction only redefines one key at a time. To redefine all four keys, you must send the KY instruction four times, once for each key.

The key definitions are only enabled while the plotter is in keyboard mode. In other words, if you send a KY instruction followed by a WD instruction, pressing a redefined function key will cause the new function to be performed. However, if you send a KY instruction but do not send a WD instruction, then pressing a "redefined" function key will cause the normal, displayed function to be performed instead of the redefined function.

NOTE: While a function key is redefined by the KY instruction, the OK instruction cannot detect whether that key has been pressed. The ESC.O instruction can still detect that a function key has been pressed, but it cannot confirm *which* key was pressed. ■

ManualsPlus.com

The KY instruction remains in effect until another KY instruction redefines or cancels the same function key, or the plotter is initialized.

The following table summarizes the possible error conditions or unexpected results that you might observe with the KY instruction.

Condition	Error	Plotter Response
no parameters	none	cancels all function keys; returns them to normal operating state
1 parameter	none	cancels the function key corresponding to the parameter given
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction

Example — Defining Function Keys with the KY Instruction

The following program is very simple and is intended only to show you the concept of how you might use redefined function keys in an interactive program. You could incorporate a program such as this into a larger plotting program.

First, this program uses the KY instruction to define key 1 to perform the P1 function, key 2 to perform the P2 function, and key 3 to perform the escape function. Key 4 is left in its undefined state. Next, the WD instruction displays labels for each key (blanks are included so that each label will be centered under the correct key). Finally, a loop is performed with the OG instruction to poll for the escape function being activated (the escape function is activated when key 3 has been pressed). When the OG instruction detects this, it automatically begins flush mode, which discards incoming HP-GL data.

As you can see in line 50, the OG instruction provides two pieces of information: the group count number (C), and a number that indicates whether the escape function has been activated (E). The group count number is most useful in spooling applications (typically in an RS-232-C configuration), so it is ignored in this program. However, the "escape function" number *is* checked in this program. If it is −1 (E = −1), the escape function has been activated, so the program does the following: sends the end flush mode instruction, ESC.U, cancels all function key definitions, and displays the message "PROGRAM TERMINATED." Otherwise, the loop continues by sending another OG instruction. (The OG instruction is described later in this chapter, and the ESC.U instruction is described in Chapter 14.)

ManualsPlus.com

Note that the CHR$ function is used to send the ASCII character **ESC** to the plotter as part of the ESC.U instruction. Also, the INPUT #1 statement is used to read the plotter's ouput response (refer to Chapter 13 for information on reading plotter output.)

```
10  'Insert configuration statement here
20  PRINT #1, "IN;KY1,4;KY2,5;KY3,12;"
30  PRINT #1, "WD    P1    P2      ESC"+CHR$(3)
40  PRINT #1, "OG;"
50  INPUT #1, C,E
60  IF E=-1 THEN PRINT #1, CHR$(27)+".U;KY;WD    PR
                 OGRAM        TERMINATED"+CHR$(3)
                 ELSE GOTO 40
70  END
```

Now load paper and run the program (although this program does not cause any plotting, paper must be loaded in order for it to run). Now press function keys 1 and 2, and observe that the pen holder moves to P1 and P2. When you want to stop the program, press key 3, which causes an escape.

The Output Key Instruction, OK

USES: The OK instruction outputs a number that indicates whether one of the front-panel function keys has been pressed. Use this instruction in conjunction with the WD instruction when designing interactive programs.

SYNTAX: *OK term*

RESPONSE: key pressed [TERM]

EXPLANATION: After an OK instruction is received, the plotter outputs an integer between 0 and 4, followed by the output terminator. The number indicates which key has been pressed, as follows. (Refer to the KY instruction, described previously in this chapter, for an illustration of the function keys and their associated numbers.)

0 No key has been pressed, the plotter is not in keyboard mode, or the key has been redefined
1 Function key 1 has been pressed
2 Function key 2 has been pressed
3 Function key 3 has been pressed
4 Function key 4 has been pressed

When a function key is pressed, bit 9 in the extended status word is set and the corresponding key number is saved by the plotter. The plotter does not sense any subsequent keys being pressed, since the bit has

ManualsPlus.com

already been set. Therefore, when the plotter receives the OK instruction, it outputs only the first key pressed. Upon completion of the OK instruction, bit 9 is cleared.

NOTE: If a function key has been redefined by the KY instruction, the OK instruction cannot detect when that key has been pressed. If you redefine the function keys, you can instead use the ESC.O instruction to detect when a key has been pressed. However, ESC.O does not determine *which* key has been pressed. ■

After you send the OK instruction, your program should read the plotter's output response. Refer to Hints for Obtaining Plotter Output Responses in Chapter 13.

The following table summarizes the possible error conditions or unexpected results that you might observe with the OK instruction.

Condition	Error	Plotter Response
no parameters	none	outputs function key information
plotter not in keyboard mode (WD instruction not in effect)	none	outputs a 0 (zero)
function key(s) redefined (KY instruction in effect)	none	outputs a 0 (zero)
1 or more parameters	2	outputs function key information

Example — Using the Function Keys with the OK and WD Instructions

The following program is a simple example that shows one way you can use the WD and OK instructions together in an interactive program. Typically, you would follow line 80 with instructions for drawing a plot. However, you can also run this program as shown. Remember that you must load paper in the plotter before running the program, even though nothing will be drawn on the paper.

First, this program uses the WD instruction to place a prompt in the display; blank spaces are included to space the prompt. The prompt tells the operator to be sure pens are loaded in the carousel, and to press a function key when he has performed this action. Then a loop in the program repeatedly sends the OK instruction to determine when a function key has been pressed. (In this program, it does not matter which key has been pressed.) When a key has been pressed (the plotter outputs a number between 1 and 4 inclusively), the program could continue with its

ManualsPlus.com

plot. To show when this occurs, this program displays the phrase "PENS ARE LOADED" and halts. The extra blank spaces are included in this message (line 80) in order to erase the characters from the previous message.

```
10  'Insert configuration statement here
20  PRINT #1, "IN;"
30  PRINT #1, "WD LOAD PENS   AND PRESS KEY"+CHR$(3)
40  PRINT #1, "OK;"
50  INPUT #1, K
60  IF K<1 THEN 40
70  IF K>4 THEN 40
80  PRINT #1, "WDPENS ARE LOADED            "+CHR$(3)
90  END
```


The Group Count Instruction, GC

USES: The GC instruction assigns a number known as the "group count," which is the number that will be output by the OG instruction. The group count is an arbitrary number that you assign at the beginning of a data block in spooling applications (typically in an RS-232-C configuration with a mainframe computer). Use this instruction in conjunction with the OG instruction to monitor the successful transfer of data blocks.

SYNTAX: *GC* count number *term*
or
GC *term*

Parameter	Format	Range	Default
count number	integer	-2^{23} to $2^{23}-1$	0

EXPLANATION: A GC instruction without parameters (GC;) sets a default group count of 0. Any other parameter sets the group count to the specified number.

The GC instruction remains in effect until another GC instruction is executed, or the plotter is initialized.

The following table summarizes the possible error conditions or unexpected results that you might observe with the GC instruction.

Condition	Error	Plotter Response
no parameters	none	sets group count to 0
more than 1 parameter	2	executes first parameter
parameter out-of-range	3	ignores instruction

ManualsPlus.com

The Output Group Count Instruction, OG

USES: The OG instruction outputs information about the current group count (data block number) and whether the escape function has been activated. Use this instruction at the end of a data block in spooling applications, where it is important to know the current data block number, and whether the data block has been transferred.

SYNTAX: *OG term*

RESPONSE: count number, escape status [TERM]

EXPLANATION: After an OG instruction is received, the plotter outputs two ASCII integers, separated by a comma and followed by the output terminator. The meanings of the integers follow.

1. Count Number. The count number is the number defined in the most recent GC instruction (or 0 if the plotter has been initialized or if there has not been a GC instruction).

2. Escape Status. The escape status provides information regarding whether the escape function has been activated. For the function to be activated, either the normal front-panel **ESCAPE** function key must be pressed, or a function key redefined by the KY instruction to perform the escape function must be pressed.

   The plotter outputs either 0 or −1, as follows:

   0 The escape function has not been activated since the last OG instruction (or since the plotter was initialized).

   −1 The escape function has been activated since the last OG instruction (or since the plotter was initialized).

If the escape function has been activated (the plotter outputs −1), the OG instruction causes the plotter to enter flush mode. In flush mode, the I/O buffer is cleared and all incoming HP-GL instructions are ignored by the plotter. The plotter remains in flush mode until it receives an ESC.U instruction, which causes the plotter to interpret HP-GL instructions again. The end flush mode instruction, ESC.U, is described in Chapter 14. For an example using the OG and ESC.U instructions, refer to The Define Key Instruction, KY, described earlier in this chapter.

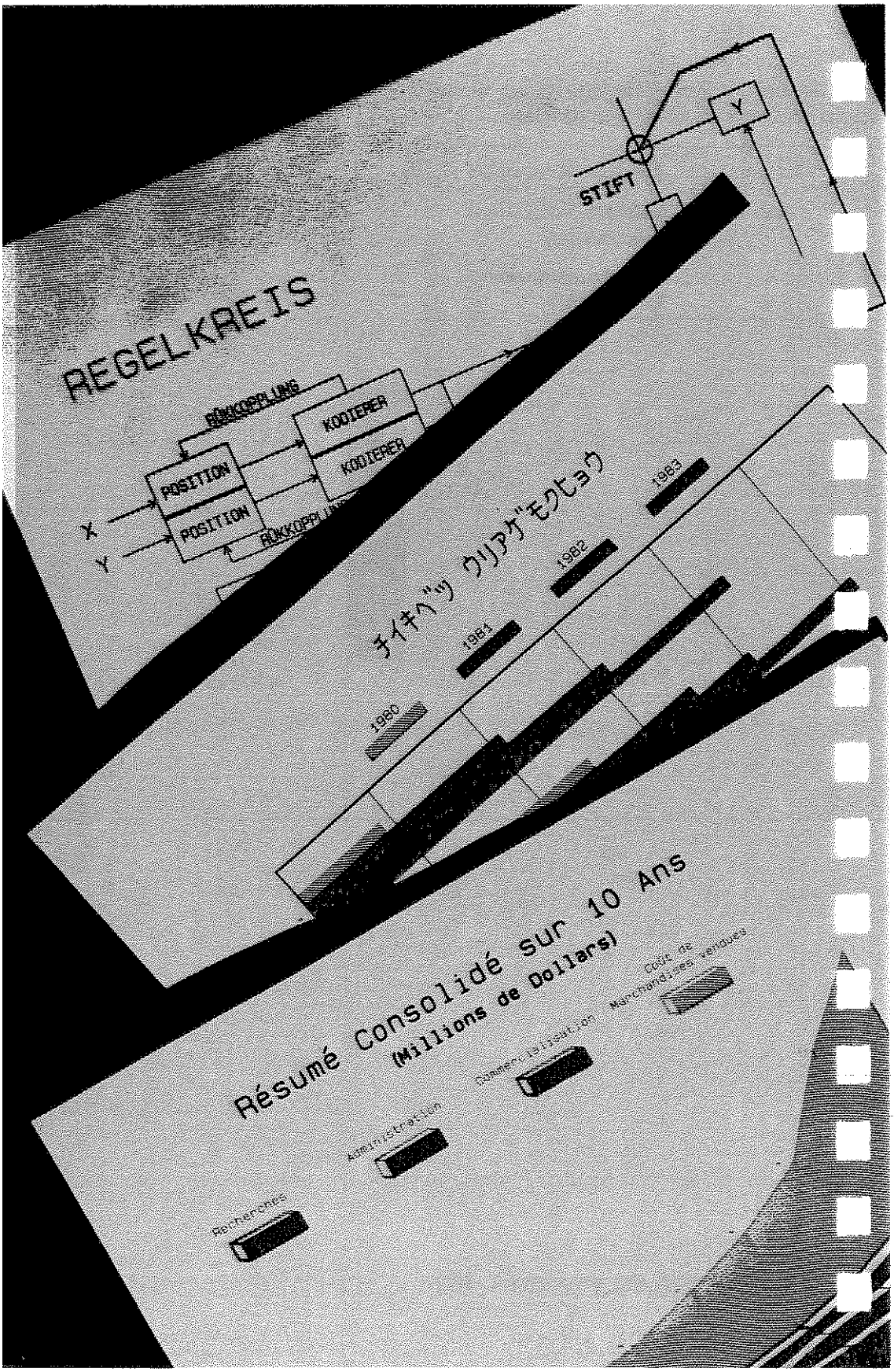
After you send the OG instruction, your program should read the plotter's output response. Refer to Hints for Obtaining Plotter Output Responses in Chapter 13.

ManualsPlus.com

The following table summarizes the possible error conditions or unexpected results that you might observe with the OG instruction.

Condition	Error	Plotter Response
no parameters	none	outputs group count information
1 or more parameters	2	outputs group count information

ManualsPlus.com



ManualsPlus.com

Chapter 11
Labeling with Alternate Character Sets and Designing Characters

What You'll Learn in This Chapter

In this chapter you will learn how to designate and select any of 20 character sets in two fonts (fixed-space and variable-space). Once you have selected a character set, you can use the labeling instructions presented in Chapter 7 to plot with the new characters. You will also learn that the plotter can label in four character selection modes; there are two HP modes and two ISO* modes. Usually you can use the default HP mode. However, if you label in many international languages, you might find that one of the ISO modes is most flexible. Finally, you will learn how to design your own character, or even your own complete character set.

HP-GL Instructions Covered

CS The Designate Standard Character Set Instruction
CA The Designate Alternate Character Set Instruction
SS The Select Standard Character Set Instruction
SA The Select Alternate Character Set Instruction
CM The Character Selection Mode Instruction
DS The Designate Character Set into Slot Instruction
IV The Invoke Character Slot Instruction
CC The Character Chord Angle Instruction
UC The User-Defined Character Instruction
DL The Define Downloadable Character Instruction

Terms You Should Understand

Character Set — a group of characters, each of which is defined by a unique ASCII decimal code. Typically, a character set contains related characters, such as a character set composed of math symbols, or a Swedish character set. The HP 7550 plotter has 20 character sets, each of which can be drawn in a fixed-space font or a variable-space font.

*International Standards Organization

ManualsPlus.com

Character Selection Mode — one of four methods implemented by the plotter that defines how you can designate and select character sets for labeling. You can use the default HP 7-bit compatibility mode, the HP 8-bit mode, the ISO 7-bit mode, or the ISO 8-bit mode.

Designate — to establish one of the plotter's character sets as a set that can be selected later for use in a labeling instruction.

Select (Invoke) — to choose one of the designated character sets as the set to be used in a labeling instruction.

Download — to store information in the plotter's buffer (memory) so that it can be accessed later.

Plotter Character Sets

The plotter has 20 character sets, each available in two fonts. In addition to these sets, you can define characters for a set known as the "downloadable" character set. This set is explained in more detail under The Define Downloadable Character Instruction, DL, at the end of this chapter.

Note that "font" simply denotes a style of lettering, which is described next. For more information on how spacing is interpreted in both fonts, refer to Adjusting Character Size, Spacing, and Position in Chapter 7.

Fixed-space font — Characters all occupy an equal horizontal space, and each character is always drawn using a fixed number of vectors.

Variable-space font — Characters each occupy a proportional horizontal space determined by the character's width. Also, the characters have a contour smoothness that is programmable (with the CC instruction).

Although the same set is available in two fonts, technically each font must be specified as a separate set. Thus, ANSI ASCII is set 0 for the fixed-space font, and set 10 for the variable-space font. In the following table, each of the first two columns represents a font. The character sets on the same line (such as 0 and 10, or 8 and 18) have the same characters for each decimal code.

ManualsPlus.com

Character Set Number		Character Set Name	ISO Registration Number
Fixed-space	Variable-space		
0	10	ANSI ASCII	006
1	11	HP 9825 HPL Character Set	—
2	12	French/German	—
3	13	Scandinavian	—
4	14	Spanish/Latin American	—
5	15	Special Symbols	—
6	16	JIS ASCII	014
7	17	Roman Extensions	—
8	18	Katakana	013
9	19	ISO IRV (International Reference Version)	002
30	40	ISO Swedish	010
31	41	ISO Swedish for Names	011
32	42	ISO Norwegian, Version 1	060
33	43	ISO German	021
34	44	ISO French	025
35	45	ISO British	004
36	46	ISO Italian	015
37	47	ISO Spanish	017
38	48	ISO Portuguese	016
39	49	ISO Norwegian, Version 2	061
−1	—	Downloadable	—

All sets except Special Symbols (sets 5 and 15), Roman Extensions (sets 7 and 17), and Katakana (sets 8 and 18) draw identical uppercase and lowercase letters and numbers. The differences between the character sets are in the *additional* characters that are needed for a certain language, for example the ä in the German sets 33 and 43.

Character sets 0 and 10 are shown on the next page. Notice the difference in the total space required to plot the sets; the variable-space font (10) takes less space. Compare also the variation in the smoothness of the characters between the fixed-space font and the variable-space font. The characters in all of the character sets are shown in a table in Appendix A.

ManualsPlus.com

```
                CHARACTER SET 0
! "#$%&' () *+, -./0123456789: ; <=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ [\] ^_`
abcdefghijklmnopqrstuvwxyz {| } ~

                CHARACTER SET 10
!"#$%&'()*+,-./0123456789:;<=>?@
ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
```

Character Selection Modes

The plotter can generate characters in any of four different modes. The default is HP 7-bit compatibility mode. With this mode, you can designate a standard set and an alternate set using the CS and CA instructions, respectively. You can then label with both character sets either by selecting the desired set with the SS and SA instructions, or by using a technique called shift-in/shift-out. These two methods are described in the next section, titled Labeling with Standard and Alternate Character Sets. Refer also to the descriptions of the CS, CA, SS, and SA instructions, which follow that section. When default conditions are established, the plotter automatically sets both the standard and alternate character sets to ANSI ASCII English, character set 0 (fixed-space).

You can also use any of three other character selection modes: HP 8-bit, ISO 7-bit, and ISO 8-bit. In each of these modes, you can label in standard and alternate sets as described above. However, you have further options for accessing up to four character sets (instead of two) at a given time. To determine whether you wish to use one of these modes, refer to Choosing Other Character Selection Modes later in this chapter.

Labeling with Standard and Alternate Character Sets

If you always intend to label with the default fixed-space set, ANSI ASCII, you do not need to execute any of the CS, CA, SS, or SA instructions before labeling. However, if you intend to use a different set (e.g., the variable-space ANSI ASCII) or additional sets, you will find that these instructions are very useful. This section describes methods for using these instructions and provides examples. The syntax of each instruction is described in detail following this section. The methods in this section are valid for any character selection mode. However, if you

ManualsPlus.com

intend to use a mode other than the default HP 7-bit mode, refer also to Choosing Other Character Selection Modes and to the CM, DS, and IV instructions later in this chapter.

Selecting Sets with the SS and SA Instructions

The principles for labeling with different character sets are very simple. Just remember that the designate character set instructions, CS and CA, tell the plotter which of the character sets will be wanted for plotting, whereas the select character set instructions, SS and SA, tell the plotter which of the two designated sets to use for subsequent labeling. Follow these steps for designating and selecting character sets.

1. Designate the standard and alternate character sets that you intend to use by executing the CS and CA instructions.

   When initialized or set to default conditions, the plotter assigns both the standard and alternate sets to be set 0. Therefore, you only need to execute a CS or CA instruction if you want to designate a character set *other than* set 0.

2. Select either the designated standard set or the designated alternate set as the set to be used for subsequent labeling by executing an SS or SA instruction.

   When initialized or set to default conditions, the plotter assumes you are labeling with the standard set. Therefore, if you want to begin by labeling with the standard set, you do *not* need to execute an SS instruction. However, if you want to begin with the alternate set, execute SA. Throughout your program, you can "toggle" between the sets by executing SS or SA, as appropriate.

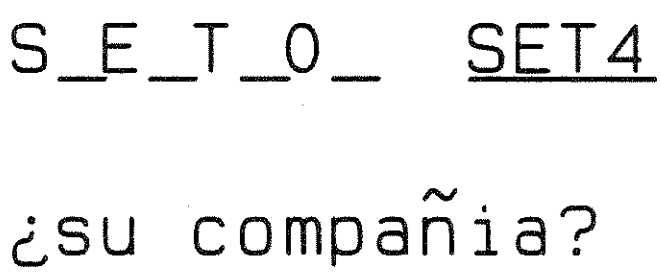
   While the standard set is selected, you can change the designated standard set with the CS instruction, and subsequent labeling will automatically be drawn in the new set (you don't have to select the set again with the SS instruction). Similarly, while the alternate set is selected, you can change the designated alternate set with the CA instruction, and subsequent labeling will automatically be drawn in the new set (you don't have to select the set again with the SA instruction).

To label characters in any character set, you can either use a computer language-dependent function such as CHR$ to enter the equivalent decimal code, or you can enter the ANSI ASCII equivalent character from your keyboard. For example, to cause the plotter to label the character "½" in set 7, you can either use "x" (the ASCII equivalent character) from the keyboard, or CHR$(120) (the equivalent decimal code). A full table of ASCII decimal codes and characters appears in Appendix A.

ManualsPlus.com

The following instructions illustrate how to designate, select, and label with standard and alternate character sets. Notice that the label string in the LB instruction shows the character that appears on the computer keyboard, unless the character is not available on the keyboard; in this case, the CHR$ function is used to access the character. (Refer to How to Send the Label Terminator and Other Nonprinting Characters under the LB instruction in Chapter 7 for more information on using the CHR$ function.) Also, notice that the "SET 4" label uses the underscore character in set 4, which automatically backspaces before it is drawn.

```
"IN;SP1;"
"PA3000,3000;CS0;CA4;"
"SS;LBS_E_T_0_"+CHR$(3)
"SA;LB  S_E_T_4_"+CHR$(3)
"CP-14,-2;LB#su compan"+CHR$(124)+"ia?"+CHR$(3)
```



Selecting Sets with the Shift-in/Shift-out Control Characters

You might find that you need to access a different character set in the middle of a label string. Using SS and SA to toggle between sets can sometimes be inefficient in this case. Instead, you can use two control characters, the shift-in (SI, decimal code 15) and the shift-out (SO, decimal code 14), to toggle between sets. The shift-in character shifts to the standard set designated by the CS instruction, whereas the shift-out character shifts to the alternate set designated by the CA instruction.

The following example shows how to use the shift-in/shift-out characters. On HP Series 200 computers, shift-in is accessed by pressing CTRL and O simultaneously, while shift-out is accessed by pressing CTRL and N simultaneously. If you cannot access these characters from your keyboard, use the CHR$ (or equivalent) function to specify the equivalent decimal code. The instructions in this example plot the same label twice, once with the equivalent keyboard characters and once with the CHR$ function.

NOTE: Although the instructions that use CHR$ are printed on two lines to fit on the page, you should send them to the plotter together as one string. ■

ManualsPlus.com

```
"IN;SP1;"
"PA300,300;CS30;CA7;SI.4,.6;"
"SS;LB3"+CHR$(14)+"x"+CHR$(15)+"-5 JR"+CHR$(3)
"CP2,0;"
"LB3"+CHR$(14)+CHR$(120)+CHR$(15)+"-5 "+CHR$(93)+
  "R"+CHR$(3)
```

The Designate Standard Character Set Instruction, CS

USES: The CS instruction designates a character set as the standard character set to be used when labeling. Use this instruction to change the standard set to one appropriate to your application. This instruction is particularly useful when plotting in a language other than English.

SYNTAX: *CS* set *term*
or
CS *term*

Parameter	Format	Range	Default
set	integer	−1, 0–19, 30–49	0

EXPLANATION: The character set designated by the CS instruction is used for all labeling operations when the standard set is selected by the SS or IV instruction, or by the control character shift-in (decimal code 15) in a label string. A CS instruction without parameters (CS;) establishes the default character set 0 (ANSI ASCII English). The CS instruction remains in effect until another CS or DS instruction is executed, or the plotter is initialized or set to default conditions.

If you execute a CS instruction while the standard set is selected (e.g., an SS instruction has been executed), subsequent labeling will be done with the set designated by the new CS instruction.

However, if you execute a CS instruction while the alternate set is selected (e.g., an SA instruction has been executed), subsequent labeling will not be done with the new standard set until it is selected with an SS or IV instruction.

For an example using the CS instruction, refer to Labeling with Standard and Alternate Character Sets earlier in this chapter.

The table on the following page summarizes the possible error conditions or unexpected results that you might observe with the CS instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	designates character set 0
more than 1 parameter	2	executes first parameter
parameter out-of-range	5	ignores instruction

The Designate Alternate Character Set Instruction, CA

USES: The CA instruction designates one of the character sets as the alternate character set to be used in labeling instructions. Use this instruction to provide an additional character set that you can easily access in a program, especially when a single label contains characters found in two different sets.

SYNTAX: *CA* set *term*
or
CA *term*

Parameter	Format	Range	Default
set	integer	−1, 0–19, 30–49	0

EXPLANATION: The character set designated by the CA instruction is used for all labeling operations when the alternate set is selected by the SA or IV instruction, or by the control character shift-out (decimal code 14) in a label string. A CA instruction without parameters (CA;) establishes the default character set 0 (ANSI ASCII English). The CA instruction remains in effect until another CA or DS instruction is executed, or the plotter is initialized or set to default conditions.

If you execute a CA instruction while the alternate set is selected (e.g., an SA instruction has been executed), subsequent labeling will be done with the set designated by the new CA instruction.

However, if you execute a CA instruction while the standard set is selected (e.g., an SS instruction has been executed), subsequent labeling will not be done with the new alternate set until it is selected with an SA or IV instruction.

For an example using the CA instruction, refer to Labeling with Standard and Alternate Character Sets earlier in this chapter.

The following table summarizes the possible error conditions or unexpected results that you might observe with the CA instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	designates character set 0
more than 1 parameter	2	executes first parameter
parameter out-of-range	5	ignores instruction

The Select Standard Character Set Instruction, SS

USES: The SS instruction selects the standard set designated by the CS or DS instruction as the character set to be used for subsequent labeling. Use this instruction to shift from the currently selected alternate set to the currently designated standard set in order to access characters in another set.

SYNTAX: *SS term*

EXPLANATION: The SS instruction tells the plotter to draw subsequent labeling instructions using characters from the character set designated by the CSn; or DS0,n; instructions. The SS instruction is equivalent to using the control character "shift-in" (decimal code 15) within a label string.

The SS instruction is in effect when the plotter is initialized or set to default conditions. The SS instruction remains in effect until an SA or IV instruction is executed.

For an example using the SS instruction, refer to Labeling with Standard and Alternate Character Sets earlier in this chapter.

The following table summarizes the possible error conditions or unexpected results that you might observe with the SS instruction.

Condition	Error	Plotter Response
no parameters	none	selects standard set
1 or more parameters	2	selects standard set

The Select Alternate Character Set Instruction, SA

USES: The SA instruction selects the alternate set designated by the CA or DS instruction as the character set to be used for subsequent labeling. Use this instruction to shift from the currently selected standard set to the currently designated alternate set in order to access characters in another set.

ManualsPlus.com

SYNTAX: *SA term*

EXPLANATION: The SA instruction tells the plotter to draw subsequent labeling instructions using characters from the character set designated by the CA n ; or DS 1 , n ; instructions. The SA instruction is equivalent to using the control character "shift-out" (decimal code 14) within a label string.

The SA instruction remains in effect until an SS or IV instruction is executed, or the plotter is initialized or set to default conditions.

For an example using the SA instruction, refer to Labeling with Standard and Alternate Character Sets earlier in this chapter.

The following table summarizes the possible error conditions or unexpected results that you might observe with the SA instruction.

Condition	Error	Plotter Response
no parameters	none	selects alternate set
1 or more parameters	2	selects alternate set

Choosing Other Character Selection Modes

The default character mode on all HP plotters is the HP 7-bit compatibility mode. In this mode, you can access any character set using the CS, SS, CA, and SA instructions, which were presented earlier in this chapter. You can also use the shift-in and shift-out control characters in a label string to access the designated standard and alternate sets, as described earlier.

The HP 7550 plotter also offers three additional character modes: the HP 8-bit mode, the ISO 7-bit mode, and the ISO 8-bit mode. The HP 8-bit mode is a tool for converting European software to HP systems software and will be of use primarily if you are using new HP systems software in a foreign language environment. The ISO modes permit more flexibility when you frequently want to access more character sets, yet conserve buffer space.

NOTE: If the default HP 7-bit mode is adequate for your labeling needs, you do not need to read this section, nor the explanations of the CM, DS, and IV instructions that follow.

If your computer system is operating with parity, you must use one of the 7-bit modes. If you use a 7-bit mode, be sure the front-panel **PARITY** function key is set to **7-BIT OFF** or **8-BIT OFF**. To use the 8-bit modes, be sure your computer system is operating with no parity, and the plotter's front-panel **PARITY** function key is set to one of the following: **8-BIT OFF**, **8-BIT ODD**, **8-BIT EVEN**, or **7-BIT OFF**. ■

ManualsPlus.com

The first three portions of this section discuss background information common to the character modes, specifically: the in-use code table, control characters, and fallback mode. The remaining portions of this section discuss each of the character selection modes. These discussions assume that you understand the basic principles of designating and selecting characters sets with the CS, CA, SS, and SA instructions (refer to Labeling with Standard and Alternate Character Sets, earlier in this chapter). These discussions also assume that you understand the sequence of instructions to use when labeling in the alternative character modes. The steps for selecting a character mode and then labeling follow:

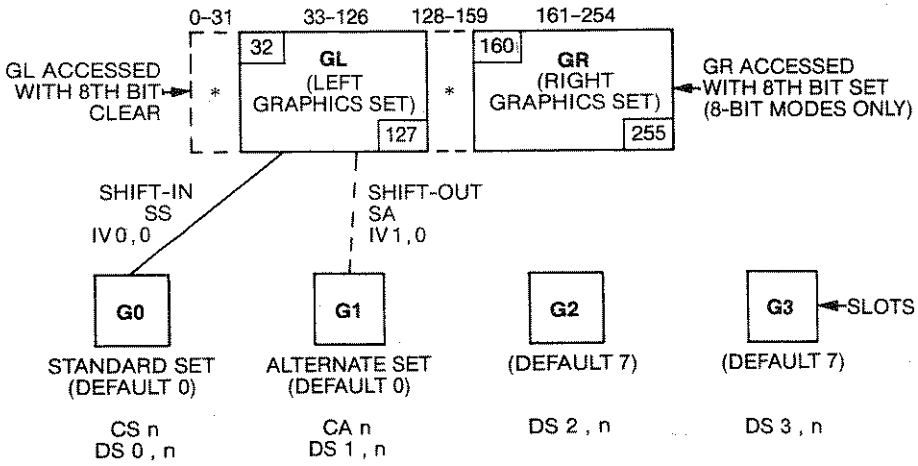
1. First execute the character selection mode instruction, CM, to establish one of the modes.
2. Then execute the designate character set into slot instruction, DS, to designate character sets (similar to the CS and CA instructions).
3. Finally, execute the invoke character slot instruction, IV, to select the set to be used for labeling (similar to the SS and SA instructions).
4. Now you can execute labeling instructions, and change set selections or designations as required.

The specific details of the syntax of the CM, DS, and IV instructions are presented following the discussions of the individual character selection modes.

The In-Use Code Table

Each character selection mode makes use of the "in-use code table," shown on the next page. This table presents the general concept of how the plotter selects character sets. In other words, this code table illustrates the underlying architecture for the selection of any character set in any mode. The descriptions of each mode later in this section present specific illustrations of how this table is interpreted for that particular mode.

ManualsPlus.com



*Codes 0–31 are control characters for all character sets, whether in GL or GR. Codes 128–159 are control characters for character sets in GR. For a list of recognized and ignored codes, refer to the section titled Control Characters.

G0 through G3 are the graphics slots. These slots designate which of the plotter's character sets may be selected (invoked into GL and GR) for labeling. When the plotter is initialized or set to default conditions, both G0 and G1 designate set 0, and both G2 and G3 designate set 7. Use the DS instruction to designate different character sets into these slots. (The CS and CA instructions are a subset of the DS instruction. If you are only interested in designating sets into slots G0 and G1, you can use the CS and CA instructions, respectively.)

GL and GR are the left graphics set and right graphics set, respectively. They contain the currently selected (invoked) character sets from the slots; all labeling instructions use the sets that currently reside in GL and GR. To place a character set into GL or GR, you would first use the DS instruction to designate character sets into the slots as described above. Then you would invoke one of the slots into GL or GR with the IV instruction. You can change the sets that reside in GL and GR by invoking different slots with new IV instructions. (The SS and SA instructions are a subset of the IV instruction. If you are only interested in invoking slots G0 and G1 into GL, you can use the SS and SA instructions, respectively. Depending on which mode you are using, you can also invoke slots by embedding shift-in, shift-out, and locking single shifts within label strings. These are discussed separately later with each mode.)

There are some restrictions on when the character sets residing in GL and GR can be accessed in labeling instructions. These restrictions depend on which character selection mode is in effect: both 7-bit modes can only access character sets in GL, whereas both 8-bit modes can access

ManualsPlus.com

character sets in both GL and GR. The following paragraphs describe how character sets are "stored" in GL and GR, depending on the mode.

- Both 7-bit modes can access a 128-character set that is contained in GL. Technically, only the printing characters (33–126) reside in GL. This is because the first 32 characters (0–31) of any set, regardless of mode, are nonprinting control characters. Characters 32 and 127 are also considered to be part of GL, although they are both nonprinting characters (32 is a space and 127 is ignored).

  To print a character residing in GL, either use the character on your keyboard that represents the associated decimal code, or use the CHR$ function (or equivalent for your computer) to specify the actual decimal code. For example, to print the character Æ (decimal 91) in set 32, you could enter [in the label string. Or, you could use the CHR$(91) function. For more examples, refer to the section earlier in this chapter, titled Labeling with Standard and Alternate Character Sets. Appendix A contains a character set table with the decimal codes of all characters.

- Both 8-bit modes can access two 128-character sets that are contained in GL and GR. In effect, you are actually accessing one linked 256-character set. GL still contains characters 33–126, and you can print characters from GL in the manner described above. However, GR contains characters 161–254. To print a character from GR, you must use the CHR$ function (or equivalent for your computer) and add 128 to the character's decimal code (as listed in Appendix A). Thus, if set 32 is in GR and you want to print the character Æ, you would use the CHR$(219) function (128 + 91 = 219).

  In the 8-bit modes, characters 0–31 and 128–159 are control characters that do not technically reside in GL or GR. In fact, you can use characters in the range 0–31 regardless of whether you are currently labeling from GL or GR. However, you can only use characters in the range 128–159 if you are currently labeling from GR. Control characters are further explained in the next section.

Control Characters

The plotter recognizes only the following control characters:

- backspace, decimal code 8
- half backspace, decimal code 9
- line feed, decimal code 10
- inverse line feed, decimal code 11
- carriage return, decimal code 13
- shift-out, decimal code 14
- shift-in, decimal code 15

ManualsPlus.com

In addition, in the ISO 8-bit mode, the plotter recognizes characters 142 (single-shift from slot G2) and 143 (single-shift from slot G3).

All other control characters in the ranges 0–31 and 128–159 are ignored.

NOTE: The full range of invocations described in ISO 2022.2, which uses two-character escape sequences, is *not* operative in this plotter's version of the ISO modes. The escape character (decimal code 27) is trapped by the plotter's escape sequence parser, which implements the immediately executed device-control (ESC.) instructions. Therefore, if the escape character is placed in a label instruction, any subsequent character that is not part of the escape sequence will appear as part of the printed label. ■

Fallback Mode

The fallback mode determines the action taken when the plotter receives undefined printing characters in a label string. "Undefined printing characters" refers only to undefined characters within the ranges 33–126 and 161–254. Examples of undefined characters are those areas that are blank in the Katakana and Roman Extension sets (see the character set table in Appendix A). The downloadable character set also consists entirely of undefined characters until you use DL to define one or more characters.

The plotter's default response for the fallback mode is to ignore all undefined characters. However, you can use the CM instruction to specify that the plotter draw a box (□) as a fallback character in place of an undefined character. Some graphics standards mandate that character codes in the range 33–126 (sometimes 127) which do not have graphic representation be executed with a special printing character. The fallback mode supports this mandate. The fallback mode is also useful for debugging when the disappearance of characters from a label string might be confusing.

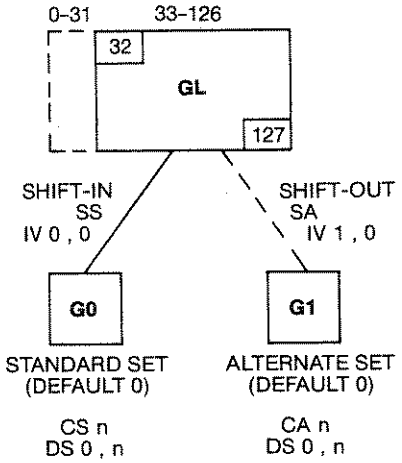
HP 7-Bit Compatibility Mode

The HP 7-bit compatibility mode is the default mode implemented on all HP plotters. To use the HP 7-bit mode, you do not need to designate sets or invoke slots with the DS and IV instructions. Instead, you can simply access any of the plotter's character sets with the CS, SS, CA, and SA instructions. You can also change character sets within a label string by using the control characters shift-out (decimal code 14) to access the alternate set, and shift-in (decimal code 15) to access the standard set. Refer to the CS, SS, CA, and SA instructions, presented earlier in this chapter, for syntax details and examples of using different character sets in this mode.

In the HP 7-bit compatibility mode, the eighth bit is always cleared and ignored. Thus, character sets can reside only in GL. All character sets

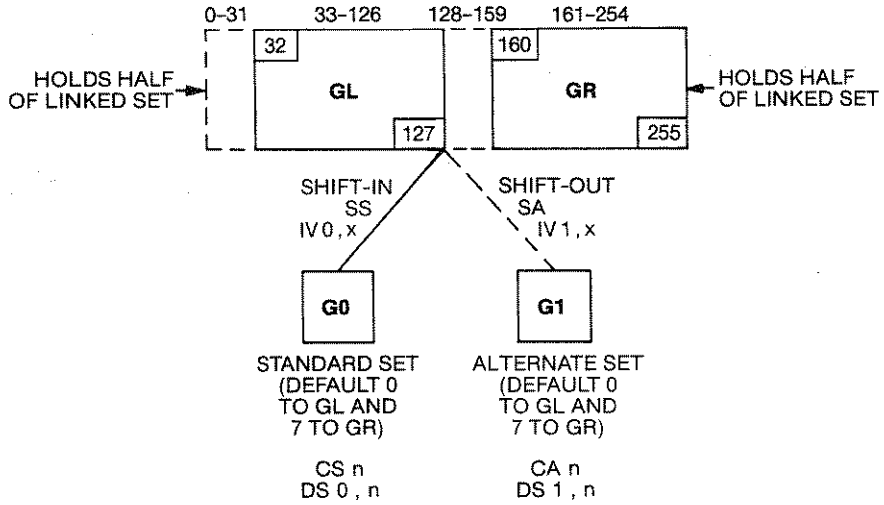
ManualsPlus.com

are composed of 128 characters. The in-use code table for the HP 7-bit compatibility mode follows.



HP 8-Bit Mode

In the HP 8-bit mode, the plotter links two 128-character sets so that they form one 256-character set. Characters 0–127 reside in GL, and characters 128–255 reside in GR. Only slots G0 and G1 are used in this mode. The in-use code table follows.



There are only two full linked sets for the HP 8-bit mode: Roman8 and Katakana8. Roman8 is composed of ANSI ASCII and Roman Extensions, and Katakana8 is composed of JIS ASCII and Katakana.

To obtain the Roman8 linked set, designate ANSI ASCII as either the standard or alternate set. Characters 33–126 will be from the ANSI ASCII

ManualsPlus.com

set, and characters 161-254 will be automatically lifted from Roman Extensions. Similarly, to obtain the Katakana8 linked set, designate JIS ASCII as either the standard or alternate set. Characters 33-126 will be from the JIS ASCII set, and characters 161-254 will be automatically lifted from Katakana. These paired relationships, shown in the table below, hold true regardless of whether fixed-space or variable-space fonts are designated.

Character Set Designated with CS, CA, or DS	GL Characters 33-126	GR Characters 161-254
ANSI ASCII (set 0 or 10)	ANSI ASCII (set 0 or 10)	Roman Extensions (set 7 or 17)
JIS ASCII (set 6 or 16)	JIS ASCII (set 6 or 16)	Katakana (set 8 or 18)

When you designate any set other than ANSI ASCII or JIS ASCII, that set is used for characters 33-126. However, in this case characters 161-254 are considered to be undefined characters and are treated according to the current fallback mode. The only way to set the eighth bit and therefore access characters in GR is to designate ANSI ASCII or JIS ASCII, as described above.

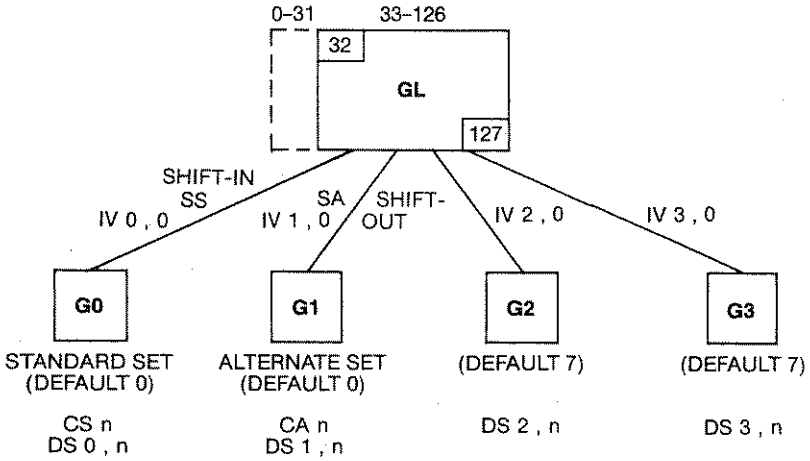
As with the HP 7-bit mode, you can change character sets within a label string by using shift-out (decimal code 14) and shift-in (decimal code 15). All invocations other than shift-in, shift-out, SS, SA, IV 0,x;, and IV 1,x; are ignored. In this mode, slots G2 and G3 cannot be designated with the DS instruction nor invoked with the IV instruction. However, G0 and G1 can be designated either with the DS instruction or the CS and CA instructions, and invoked either with the IV instruction or the SS and SA instructions.

NOTE: In order to access a character in GR, you must use a function such as CHR$ (or equivalent for your computer). To determine the decimal code to use in the CHR$ function, simply add 128 to the decimal code of the desired character. Thus, to label the character ¢ in Roman8 Extensions, you would use the CHR$(191) function (128 + 63). You can find the decimal codes of characters in Appendix A. ■

ISO 7-Bit Mode

In the IS0 7-bit mode, the eighth bit is always cleared and ignored. Thus, character sets can reside only in GL. All character sets are composed of 128 characters. When the plotter is initialized or default conditions are established, character set 0 resides in G0 and G1 and character set 7 resides in G2 and G3. The in-use code table for the ISO 7-bit mode follows.

ManualsPlus.com



The in-use code table shows which instructions may be used for designating character sets into slots, and for invoking slots into GL and GR. All invocations with the IV instruction are locking, meaning that the invoked set remains resident in GL until it is overridden with a new IV instruction or the plotter is initialized or set to default conditions. If you invoke a slot into GR, the plotter will automatically invoke the slot into GL instead. (For example, IV 3,1; which normally invokes slot 3 into GR will instead invoke slot 3 into GL.)

Example — Using the DS and IV Instructions in the ISO 7-Bit Mode

The following program illustrates the procedure for designating and invoking character sets when using the ISO 7-bit mode. Be sure the front-panel **PARITY** function key is set to **7-BIT OFF** or **8-BIT OFF**.

```
English  Français  Español
```

```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;SI.3,.4;PA100,1000;"
30  PRINT #1, "CM2;DS1,2;DS2,37;"
40  PRINT #1, "LBEnglish  "+CHR$(14)+"Fran"+CHR$(92)
                +"ais"+CHR$(3)
50  PRINT #1, "IV2;LB  Espa"+CHR$(124)+"ol"+CHR$(3)
60  PRINT #1, "SPO;"
70  END
```

Program Explanation

10 defines the plotter as the system printer — change as necessary

20 initializes the plotter; selects pen 1; establishes an absolute character size; sets the initial pen position

ManualsPlus.com

30 establishes the ISO 7-bit mode; designates character set 2 (French) into slot G1; designates character set 37 (ISO Spanish) into slot G2

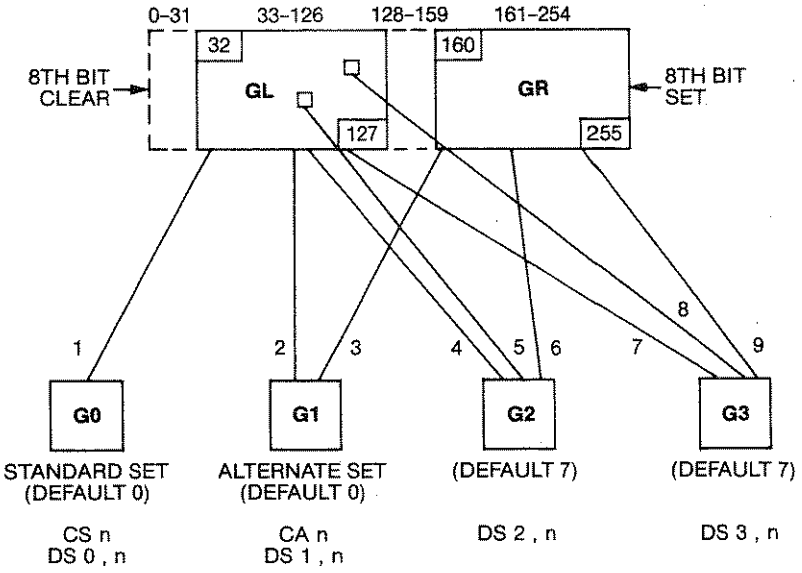
40 prints "English" with character set 0 from slot G0 (the defaults); uses shift-out (decimal code 14 or **CTRL N** on HP Series 200 computers) to invoke slot G1 and prints "Français"

50 invokes slot G2 into GL and prints "Español"

60 returns the pen to the carousel

ISO 8-Bit Mode

The ISO 8-bit mode enables you to access two character sets at once without having to use invocations and designations. You can also access four character sets in any label string with the same economy. When the plotter is initialized or default conditions are established, character set 0 resides in G0 and G1 and character set 7 resides in G2 and G3. The in-use code table for the ISO 8-bit mode follows. The numbered items refer to the possible invocations of the slots into GL or GR. The possible designations of the slots are also shown.



1 use G0 for GL (shift-in, SS, or IV 0 , 0)
2 use G1 for GL (shift-out, SA, or IV 1 , 0)
3 use G1 for GR (IV 1 , x)
4 use G2 for GL (IV 2 , 0)
5 use G2 for next GL character only (single-shift G2)
6 use G2 for GR (IV 2 , x)
7 use G3 for GL (IV 3 , 0)
8 use G3 for next GL character only (single-shift G3)
9 use G3 for GR (IV 3 , x)

ManualsPlus.com

You can invoke slots into GL using the IV, SS, or SA instructions, or by embedding shift-in (decimal code 15), shift-out (decimal code 14), and single-shift (decimal codes 142 and 143) control characters. However, the only way to invoke slots into GR is with the IV instruction.

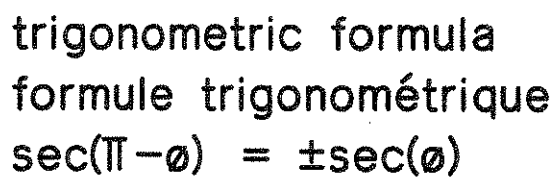
Invocations can be locking or single. All invocations with the IV instruction are locking: the invoked set remains resident in GL or GR until it is overridden with a new IV, SS, or SA instruction, or until the plotter is initialized or set to default conditions.

A single invocation is in effect for one character only. Single-shift G2 (decimal code 142) invokes slot G2, and single-shift G3 (decimal code 143) invokes slot G3. After you make either single invocation, the plotter prints the next character from the single-invoked set, then returns to the previous character set.

NOTE: In order to access the single-shift control codes or a character in GR, you must use a function such as CHR$ (or equivalent for your computer). To determine the decimal code to use in the CHR$ function, simply add 128 to the decimal code of the desired character. Thus, to label an A when set 0 is in GR, you would use the CHR$(193) function (128 + 65). You can find the decimal codes of characters in Appendix A. ■

Example — Embedding a Single-Shift in a Label String

The following example shows how to embed a single-shift character in a label string. The variable-space fonts of four character sets are used in this example: the ANSI ASCII English set (10), the French/German set (12), the Special Symbols set (15), and the Roman Extensions set (17). Be sure the front-panel **PARITY** function key is set to one of the following: **8-BIT OFF**, **8-BIT ODD**, **8-BIT EVEN**, or **7-BIT OFF**.




```
10  'Insert configuration statement here
20  PRINT #1, "IN;SP1;SI.3,.4;PA100,2000;"
30  PRINT #1, "CM3;DS0,10;DS1,12;DS2,15;DS3,17;"
40  PRINT #1, "LBtrigonometric formula"+CHR$(3)
50  PRINT #1, "CP;LB"+CHR$(14)+"formule trigonome't
      rique"+CHR$(3)
60  PRINT #1, "CP;LBsec("+CHR$(142)+"n-"+CHR$(143)+
      "V)"+CHR$(3)
70  PRINT #1, "LB = "+CHR$(142)+"psec("+CHR$(143)+"
      V)"+CHR$(3)
80  PRINT #1, "SP0;"
90  END
```

ManualsPlus.com

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 initializes the plotter; selects pen 1; establishes an absolute character size; sets the initial pen position

30 establishes the ISO 8-bit mode; designates character set 10 into slot 0; designates character set 12 into slot 1; designates character set 15 into slot 2; designates character set 17 into slot 3

40 labels "trigonometric formula" with character set 10 from the default slot G0

50 performs a carriage return and line feed; performs a shift-out (decimal code 14) to label "formule trigonométrique" with character set 12 from slot G1

60 performs carriage return and line feed; labels "sec $(\pi-\theta)$" as follows: uses decimal code 142 for a single shift to pick up "π" from character set 15 in G2 and uses decimal code 143 for a single shift to pick up "θ" from character set 17 in G3

70 labels "$\pm$ sec (θ)" as follows: uses decimal code 142 for a single shift to pick up "$\pm$" from character set 15 in slot G2 and uses decimal code 143 for a single shift to pick up "θ" from character set 17 in G3

80 returns the pen to the carousel

The Character Selection Mode Instruction, CM

USES: The CM instruction specifies the mode of character set selection and usage. Use this instruction to access one of four different character set modes: the HP 7-bit mode, the HP 8-bit mode, the ISO 7-bit mode, or the ISO 8-bit mode.

SYNTAX: *CM* switch mode (, fallback mode) *term*
or
CM *term*

Parameter	Format	Range	Default
switch mode	integer	0–3	0
fallback mode	integer	0–1	0

ManualsPlus.com

EXPLANATION: A CM instruction without parameters (CM ;) defaults to the HP 7-bit compatibility mode and fallback mode 0. When only the first parameter is given, the second parameter defaults to 0. The following paragraphs list further details of each parameter.

1. Switch Mode. The switch mode chooses which character selection mode will be used by all subsequent labeling instructions. The meanings of the four parameters follow. For more details, refer to the separate descriptions of each mode earlier in this chapter.

   0 HP 7-bit compatibility mode
   1 HP 8-bit mode
   2 ISO 2022.2 7-bit mode
   3 ISO 2022.2 8-bit mode

   NOTE: If your computer system is operating with parity, you must use one of the 7-bit modes. If you use a 7-bit mode, be sure the front-panel **PARITY** function key is set to **7-BIT OFF** or **8-BIT OFF**. To use the 8-bit modes, be sure your computer system is operating with no parity, and the plotter's **PARITY** function key is set to one of the following: **8-BIT OFF**, **8-BIT ODD**, **8-BIT EVEN**, or **7-BIT OFF**. ■

2. Fallback Mode. The fallback mode determines what the plotter does when it encounters undefined printing characters. The meanings of the two parameters follow. For more details, refer to Fallback Mode earlier in this chapter.

   0 Undefined characters are ignored (default)
   1 A box (□) is drawn in place of undefined characters

Executing a CM instruction clears the label buffer. If you execute a CM instruction that changes from the current mode to a different mode, the plotter sets default designations and invocations for the new mode (refer to the DS and IV instructions, which follow). The CM instruction remains in effect until another CM instruction is executed, or the plotter is initialized or set to default conditions.

The CM instruction determines the implementation of the designate character set into slot instruction, DS, and the invoke character slot instruction, IV. All labeling instructions (LB, BL, and PB) operate according to the character selection mode established by the CM instruction.

For examples using the CM instruction, refer to the separate descriptions of each character selection mode presented previously in this chapter.

ManualsPlus.com

The following table summarizes the possible error conditions or unexpected results that you might observe with the CM instruction.

Condition	Error	Plotter Response
no parameters	none	establishes HP 7-bit compatibility mode and ignores undefined characters
1 parameter	none	establishes specified mode and ignores undefined characters
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction

The Designate Character Set into Slot Instruction, DS

USES: The DS instruction designates up to four character sets to be immediately available for labeling instructions. Each character set is designated in one of the slots (G0–G3). Use this instruction primarily with the ISO modes for accessing multiple character sets.

SYNTAX: *DS* slot, set *term*
or
DS *term*

Parameter	Format	Range	Default
slot	integer	0–1 (HP modes) 0–3 (ISO modes)	0
set	integer	−1, 0–19, 30–49	0

EXPLANATION: A DS instruction without parameters (DS;) establishes default conditions (character set 0 in slots G0 and G1, and set 7 in slots G2 and G3). The following paragraphs list further details of each parameter.

1. Slot. The slot parameter represents the part of the in-use code table where character sets can be designated. Once it is designated, you can invoke any slot into GL or GR for use in a labeling instruction. For more details, refer to The In-Use Code Table and to the separate character selection mode descriptions earlier in this chapter. The slot parameters follow.

   0 Slot G0
   1 Slot G1
   2 Slot G2
   3 Slot G3

ManualsPlus.com

2. Set. The set parameter is the number of the character set to be designated into the slot. The character set numbers are listed under Plotter Character Sets at the beginning of this chapter, and in Appendix A.

The character selection mode instruction, CM, determines the implementation of the DS instruction, as follows. Refer to the separate descriptions of each character selection mode earlier in this chapter for examples and more details of how DS is implemented.

Character Selection Mode	Implementation of DS
HP 7-bit (default, CM 0)	DS 2, n and DS 3, n are ignored.
HP 8-bit (CM 1)	DS 2, n and DS 3, n are ignored. If DS 0, n or DS 1, n and n = 0 or 10, then set 7 or 17, respectively, is placed in G2 and G3. If DS 0, n or DS 1, n and n = 6 or 16, then set 8 or 18, respectively, is placed in G2 and G3. If DS 0, n or DS 1, n and n = −1, 1–5, 7–9, 11–15, 17–19, or 30–49, then null set is placed in G2 and G3; therefore G2 and G3 contain undefined characters.
ISO 7-bit (CM 2)	All forms of DS are accepted.
ISO 8-bit (CM 3)	All forms of DS are accepted.

Under default conditions, slots G0 and G1 each contain character set 0, and slots G2 and G3 each contain character set 7 (ignored in the HP modes). If you want to designate character sets into more than one slot, you must execute the DS instruction once for each slot. Remember that you can use the CS and CA instructions for designating character sets into slots G0 and G1, respectively.

The designation for a particular slot remains in effect until another DS (or CS or CA) instruction is executed for that slot, or the plotter is initialized or set to default conditions.

The table on the following page summarizes the possible error conditions or unexpected results that you might observe with the DS instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	designates set 0 for slots G0 and G1, and set 7 for slots G2 and G3
1 parameter	2	ignores instruction
more than 2 parameters	2	executes first 2 parameters
slot parameter out-of-range	3	ignores instruction
set parameter out-of-range	5	ignores instruction

The Invoke Character Slot Instruction, IV

USES: The IV instruction invokes a character slot (G0 through G3) into either the left or right half of the in-use code table (GL or GR, respectively). Use this instruction primarily with the ISO modes to access multiple character sets.

SYNTAX: *IV* slot (, left) *term*
or
IV *term*

Parameter	Format	Range	Default
slot	integer	0–1 (HP modes) 0–3 (ISO modes)	0
left	integer	0–1	0

EXPLANATION: An IV instruction without parameters (IV;) establishes default conditions as follows: for all modes, G0 is invoked into GL; for ISO 8-bit mode, G1 is invoked into GR. The following paragraphs list further details of each parameter.

1. Slot. The slot parameter is the number of the slot to be invoked into the left or right half of the in-use code table (GL or GR, respectively). For more details, refer to The In-Use Code Table and to the separate descriptions of the character selection modes earlier in this chapter. The values for the slot parameters follow.

   0 Slot G0
   1 Slot G1
   2 Slot G2
   3 Slot G3

ManualsPlus.com

2. Left. This parameter specifies which side of the in-use code table will receive the character set, as follows. If this parameter is omitted, the slot is invoked into GL.

    0 slot invoked into GL
    1 slot invoked into GR

The character selection mode instruction, CM, determines the implementation of the IV instruction, as follows. Refer to the separate descriptions of each character selection mode earlier in this chapter for examples and more details of how IV is implemented.

Character Selection Mode	Implementation of IV
HP 7-bit (default, CM 0)	Slots G2 and G3 cannot be invoked. Also, GR cannot be accessed explicitly. Therefore, IV 0 , x (or IV 2 , x) invokes slot G0 into GL; IV 1 , x (or IV 3 , x) invokes slot G1 into GL. Nothing is invoked into GR.
HP 8-bit (CM 1)	Same as above, except that GL and GR are linked; i.e., either the null set or the right half of a linked set is automatically invoked into GR, depending on which character set has been designated.
ISO 7-bit (CM 2)	All four slots can be invoked; however, they will automatically be invoked into GL, regardless of whether you specify GL or GR.
ISO 8-bit (CM 3)	All forms of the IV instruction are executed as specified.

NOTE: The instruction IV 0 , 0 ; or IV 0 ; is the same as SS ; or a shift-in character (decimal code 15) in the label string. All invoke the standard set from slot G0 into GL. Similarly, IV 1 , 0 ; or IV 1 ; is the same as SA ; or a shift-out character (decimal code 14) in the label string. All invoke the alternate set from slot G1 into GL. ■

All invocations with the IV instruction are locking. The invoked set remains in GL or GR until overwritten with a new instruction (IV, SS, or SA), or until the plotter is initialized or set to default conditions.

The table on the following page summarizes the possible error conditions or unexpected results that you might observe with the IV instruction.

ManualsPlus.com

Condition	Error	Plotter Response
no parameters	none	invokes G0 into GL (and if ISO 8-bit mode, G1 into GR)
1 parameter	none	invokes specified slot into GL
more than 2 parameters	2	executes first 2 parameters
parameter out-of-range	3	ignores instruction

The Character Chord Angle Instruction, CC

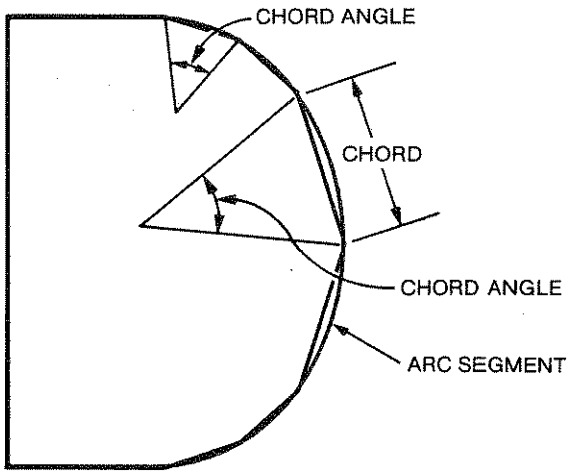
USES: The CC instruction sets the angle that determines the smoothness of characters drawn with the variable-space fonts (character sets 10–19 and 40–49). Use this instruction to reduce the number of chords used to draw characters in order to increase labeling speed. In this way, you can draw rough or preliminary plots in a minimum of time. In rare cases where characters are not smooth enough, you can also use CC to increase the number of chords to produce extremely smooth characters.

SYNTAX: *CC* chord angle *term*
or
CC *term*

Parameter	Format	Range	Default
chord angle	decimal	$\lvert -2^{23}$ to $2^{23}-1\rvert$ degrees	5 degrees

EXPLANATION: Characters in the variable-space fonts are drawn as "arc-based characters." These characters are drawn using a series of chords that each subtend a given angle. The concept is similar to the degrees mode of the chord tolerance parameter of the arc, circle, and wedge instructions (refer to Chapter 6). However, with characters, the arcs used to define a character might have different radii. Therefore, while the chord angle remains the same, the chords are different lengths because the radii of the arcs are different lengths. For example, refer to the following illustration: the D is drawn with a single chord angle of 45 degrees, but the chords are defined by arcs that have two different radii. Different characters have different numbers of radii.

ManualsPlus.com



If you specify an angle larger than ±45, the plotter uses +45 degrees for the chord angle, since larger chord angles would result in very poor character quality. If you specify an angle of 0 degrees, the plotter uses a very small chord angle instead.

A CC instruction without parameters (CC;) establishes the default of 5 degrees. The CC instruction remains in effect until another CC instruction is executed, or the plotter is initialized or set to default conditions.

The following table summarizes the possible error conditions or unexpected results that you might observe with the CC instruction.

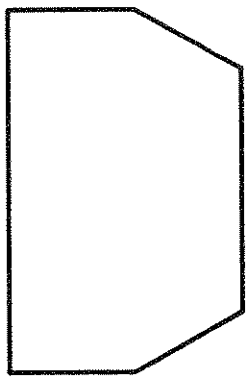
Condition	Error	Plotter Response
no parameters	none	establishes 5-degree chord angle
parameter larger than ±45	none	establishes 45-degree angle
parameter equal to 0	none	establishes very small angle
more than 1 parameter	2	executes first parameter
parameter out-of-range	3	ignores instruction

Example — Effects of Changing Character Chord Angles

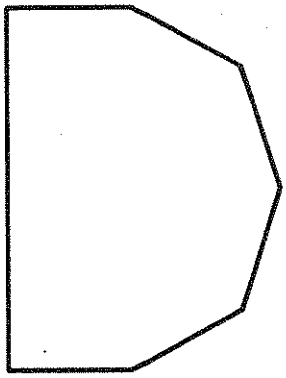
Increasing the chord angle decreases the number of chords used to construct a character, thereby decreasing the time required to draw the character and producing a rougher-looking character. Decreasing the chord angle has the opposite effect. The following program lines illustrate these effects.

ManualsPlus.com

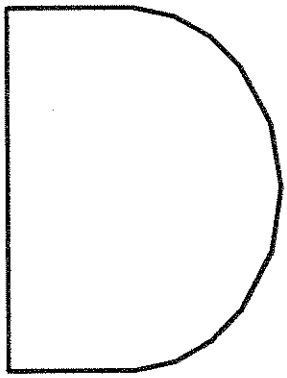
```
"IN;SP1;"
"CS10;SI3.6,4.8;PA1000,5000;"
"CC45;LBD"+CHR$(3)+"CC30;LBD"+CHR$(3)
"CP-2,-.75;CC15;LBD"+CHR$(3)+"CC5;LBD"+CHR$(3)
"SP0;"
```



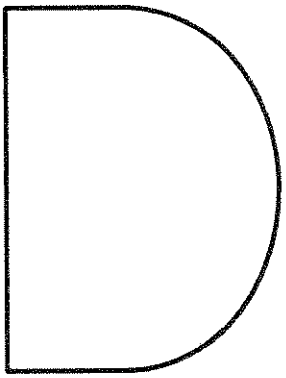
45-Degree Chord Angle



30-Degree Chord Angle



15-Degree Chord Angle



5-Degree Chord Angle

The User-Defined Character Instruction, UC

USES: The UC instruction draws a character that you design. Use this instruction to create symbols not included in the plotter's character sets, or to draw logos.

ManualsPlus.com

SYNTAX: *UC* (pen control,) X-increment, Y-increment
(, pen control) (, . . .) *term*
or
UC *term*

Parameter	Format	Range	Default
pen control	integer	**STANDARD:*** ⩾+99 = pen down ⩽−99 = pen up **ENHANCED:*** ⩾+9999 = pen down ⩽−9999 = pen up	pen up
X- and Y-increments	integer	**STANDARD:*** −98 to 98 **ENHANCED:*** −9998 to 9998 (both in primitive grid units)	none

*The ranges depend on the setting of the front-panel **STANDARD/ENHANCED** function key, as shown. The pen control parameters cannot exceed the plotter's range of -2^{23} to $2^{23}-1$.

EXPLANATION: A UC instruction without parameters (UC;) causes the pen to move to the carriage-return point. A UC instruction with parameters draws the character specified by the parameters. The parameters are interpreted as follows.

1. Pen Control. The UC instruction initially raises the pen (regardless of the current pen status). Therefore, the character will not be drawn if you do not include at least one pen down parameter. Once a pen down parameter is specified, the pen remains down for subsequent X,Y increment moves until a pen up parameter is specified or the UC instruction is terminated. You can include as many pen control parameters as you need to draw your character. Upon termination of the UC instruction, the pen is raised and moves to the next character origin, as determined by the character plot cell and any extra space set by the ES instruction. The current pen status is then restored.

2. X,Y Increments. The X,Y increment pairs represent relative positions on a primitive grid that is superimposed on the character plot cell. (The resolution of the primitive grid depends on the setting of the **STANDARD/ENHANCED** function key, and is described later in this section.)

   The UC instruction initially sets the pen position at 0,0 on the primitive grid. The pen then moves consecutively to the points defined by the X,Y increment pairs. You can include as many X,Y increment

ManualsPlus.com

pairs as you need to draw your character. The X,Y increments are interpreted as follows. All references to the right, left, up, and down are relative to the current label direction.

- The X-increment specifies the number of primitive grid units that the pen will move horizontally from the current pen position on the grid. A positive increment moves the pen to the right; a negative increment moves the pen to the left.
- The Y-increment specifies the number of primitive grid units that the pen will move vertically from the current pen position on the grid. A positive increment moves the pen up; a negative increment moves the pen down.

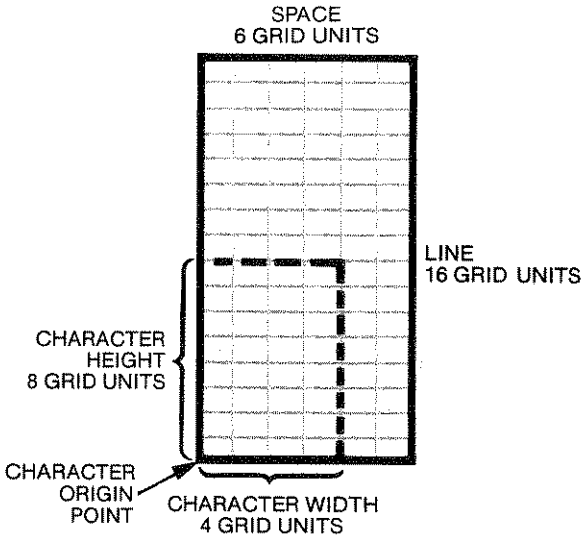
To define a character, you must specify the X,Y increments in primitive grid units. The plotter interprets the primitive grid units according to the setting of the **STANDARD/ENHANCED** function key, as shown in the next table.

Function Key	Primitive Grid Resolution of Character Plot Cell
STANDARD (either fixed-space of variable-space font currently selected)	1 space = 6 grid units 1 line = 16 grid units To match the size of other characters drawn with LB, BL, and PB instructions, restrict the character to a 4 × 8 grid.
ENHANCED (fixed space font currently selected)	1 space = 48 grid units 1 line = 64 grid units To match the size of other characters drawn with LB, BL, and PB instructions, restrict the character to a 32 × 32 grid.
ENHANCED (variable-space font currently selected)	1 space = 42 grid units 1 line = 72 grid units To match the size of other characters drawn with LB, BL, and PB instructions, restrict the character to a 28 × 36 grid.

ManualsPlus.com

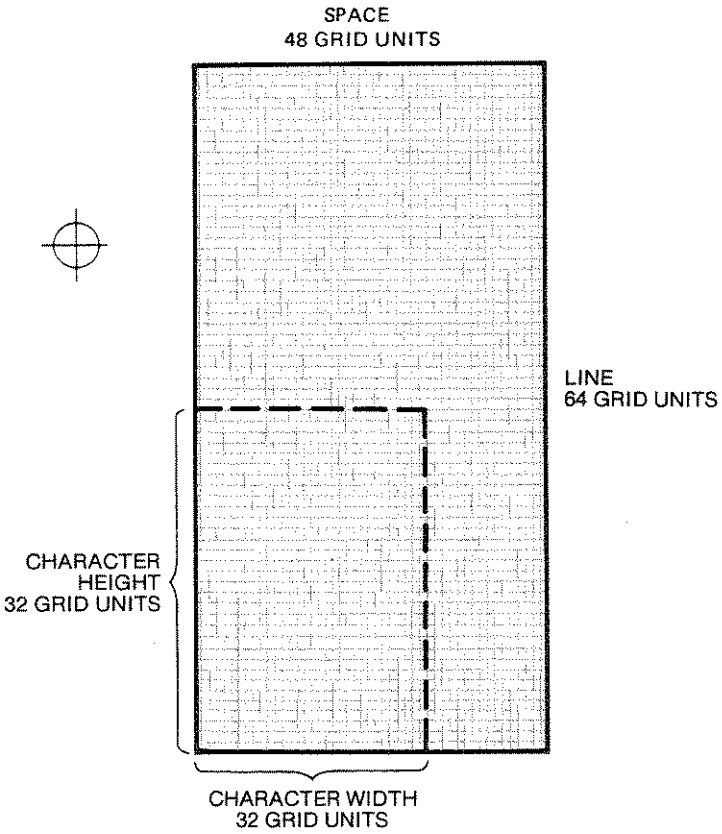
The following illustrations show the various primitive grids. Notice that the number of grid units in a character plot cell does not change. However, the size of the cell can change. That is, if you issue a character size instruction (SI or SR), the cell will always be 1.5 times the specified width and 2 times the specified height. Thus, the cell can be larger or smaller, but the number of grid units in the cell will always be the same.

You can extend your character into the area normally reserved for the space around a character, and even into the next character plot cell. This is illustrated in the example at the end of this section. However, if you want your character to be the same size as characters in the font you've selected, confine your character to the grid sizes noted in the previous table (e.g., 28 × 36 for variable-space fonts in the **ENHANCED** mode).



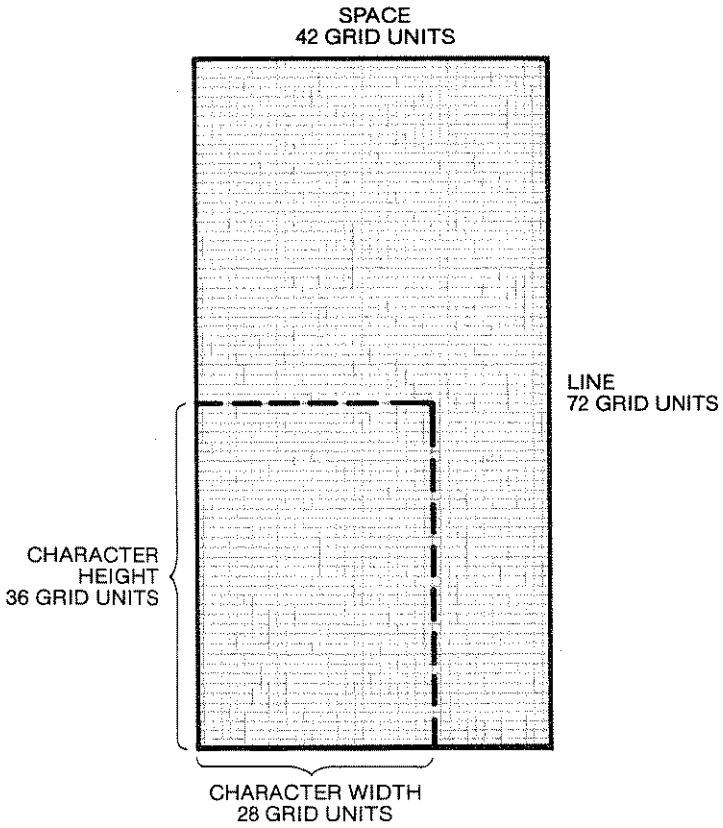
STANDARD Mode, Any Font Selected

ManualsPlus.com



ENHANCED Mode, Fixed-Space Font Selected

ManualsPlus.com



ENHANCED Mode, Variable-Space Font Selected

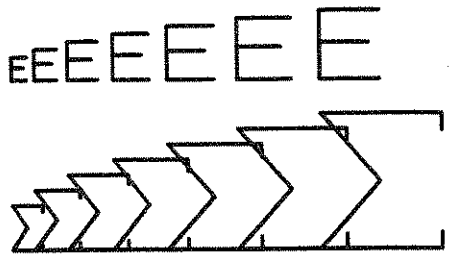
User-defined characters are drawn using the current direction (DI or DR), size (SI or SR), and slant (SL). The user-defined character is also subject to the vertical positioning component of the current LO instruction, but not to the horizontal positioning component.

As soon as the UC instruction is executed, the character defined by its parameters is drawn. To draw the character more than once, you need to put the UC instruction and parameters in a loop, as shown later in the example, or issue the UC instruction each time you want the character drawn. If you want to define a character and be able to access it repeatedly in a labeling instruction, refer to The Downloadable Character Instruction, DL, described next in this chapter.

The table on the following page summarizes the possible error conditions or unexpected results that you might observe with the UC instruction.

ManualsPlus.com

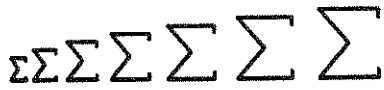
Now delete line 70 and run the program again. Notice that while the E's are drawn with normal spacing, the Σ's are partially superimposed. After drawing the user-defined character, the pen always moves to the next character plot cell measured from the original character origin. In this case, part of the Σ extends into the next character plot cell.



As mentioned in the explanation of the UC instruction, you can change the size of a user-defined character with the SI or SR instructions. However, you can also change the size by specifying different increments in the UC instruction itself. In the previous examples, the Σ was twice as high and twice as wide as the E because the Σ was drawn to the full height and more than the full width of the cell. Now reduce the Σ so that it is the same height and width as the E. To do this, change line 60 of the previous program by dividing each X,Y increment by two, as shown. The Σ is now confined to the 32 × 32 grid for characters in a fixed-space font. This is obvious when you plot the E's and Σ's together.

NOTE: Although this program line is printed on two lines to fit on this page, you should send it to the plotter as one string. ■

```
60    PRINT #1, "UC32,28,9999,0,4,-32,0,16,-16,-16,
                 -16,32,0,0,4;"
```



The Define Downloadable Character Instruction, DL

USES: The DL instruction enables you to design characters and save them in a buffer for convenient repeated use in labeling instructions.

ManualsPlus.com

You can use separate DL instructions to specify up to 94 characters, which the plotter treats as a fixed-space character set. Use the DL instruction to create characters or symbols not included in the plotter's character sets, or to create another type style.

SYNTAX: *DL* character number (, pen control), X-coordinate, Y-coordinate (,...) (, pen control) (,...) *term*
or
DL character number *term*
or
DL *term*

Parameter	Format	Range	Default
character number	integer	33–126	none
pen control	integer	−128	none
X- and Y-coordinates	integer	−127–127 primitive grid units	none

EXPLANATION: A DL instruction without parameters (DL;) clears the entire downloadable character buffer. A DL instruction with only the first parameter (character number) clears only that character from the buffer. A DL instruction with more than one parameter defines a character and places the character in the downloadable character buffer. The parameters are interpreted as follows.

1. Character Number. The first parameter of the DL instruction is always interpreted to be the character number, which is the decimal code that you assign to the character being defined. You are limited to the printing characters, decimal codes 33–126, for identifying downloadable characters. If the character number has been previously defined, the new definition overwrites the old one.
2. Pen Control. This parameter raises the pen for the next X,Y coordinate pair only. The pen control parameter is optional when it immediately follows the character number. This is because the first X,Y coordinate pair always defines a move with the pen up, regardless of the current pen status or the pen control parameter (if any). Except when preceded by the pen control parameter, all subsequent parameter pairs define a draw with the pen down.
3. X- and Y-coordinates. The X,Y coordinate pairs represent absolute positions on the primitive grid. The point 0,0 is the character origin of the character plot cell. You can place up to 255 pen control parameters and X,Y coordinates in a DL instruction.

Defining a downloadable character is similar in concept to defining a character with the UC instruction, in that parameters are expressed in terms of a primitive grid. However, note these differences:

ManualsPlus.com

- the DL instruction uses absolute coordinates (the UC instruction uses relative increments)
- in the DL instruction, the first pen movement is always with the pen up (in the UC instruction, you can place the pen down for the first movement)
- only one grid is used in the DL instruction, regardless of the setting of the **STANDARD/ENHANCED** function key or the current character font (the UC instruction uses one of three grids)

The primitive grid that the DL instruction uses has the same resolution as the grid that the UC instruction uses when a fixed-space font has been selected in the **ENHANCED** mode. (The character occupies 32×32 grid units, and the cell is 48×64 grid units.) This is also the same grid used internally for all of the plotter's fixed-space fonts. The example at the end of this section shows the primitive grid and character plot cell. Note that characters should be defined within the 32×32 area to be the same size as other fixed-space characters. However, you can define characters anywhere within a range of -127 to 127.

Before defining a character with the DL instruction, you should allocate space in the downloadable character buffer; refer to Allocating Memory for Downloadable Characters later in this section. You can define up to 94 characters, using separate DL instructions for each character. You can also define characters in any order (for example, decimal code 39 then decimal code 34, or vice versa).

When you have defined one or more downloadable characters, they reside in character set -1. Therefore, you can access a downloadable character in a labeling instruction (LB, BL, PB, or SM) in the same manner as you would access a character from any other character set. Simply designate and select (or invoke) character set -1. Note that when you access a downloadable character in the SM instruction, it will not be centered unless it was defined that way (centered within the 32×32 portion of the grid).

Because downloadable characters are drawn by labeling instructions, they are subject to the current size (SI or SR), direction (DI or DR), slant (SL), label origin (LO), and extra space (ES) settings. Also, after any downloadable character is drawn, the pen moves to the next character origin. Therefore, if you have defined a character that is larger than the normal 32×32 portion of the character plot cell, you might need to use CP, ES, PA, or PR instructions to prevent the next character from being superimposed on the oversized character. This effect is similar to what happens with user-defined characters that extend beyond the character plot cell; refer to the UC instruction for an example.

ManualsPlus.com

A DL instruction remains in effect until another DL instruction for the same decimal code is executed, a DL; instruction is executed, or the plotter is initialized or set to default conditions.

The following table summarizes the possible error conditions or unexpected results that you might observe with the DL instruction.

Condition	Error	Plotter Response
no parameter	none	clears the downloadable character buffer
1 parameter	none	clears the specified character from the downloadable character buffer
odd number of coordinates or pen control parameter with no coordinates	3	ignores instruction
more than 255 pen control, X-coordinate, and Y-coordinate parameters	2	executes first 255 parameters
parameter out-of-range	3	ignores instruction
downloadable character buffer overflow	7	erases new and previous character definition for that decimal code

Allocating Memory for Downloadable Characters

Before you define a character with the DL instruction, you must allocate space in the downloadable character buffer with the GM or ESC.T instruction. Each character requires:

- one byte for each pen control parameter
- one byte for each X-coordinate
- one byte for each Y-coordinate
- one additional byte for the byte count

The minimum amount of space required for downloadable characters is 454 bytes for overhead, plus whatever you need according to the above formula to define each character. For the gamma defined in the program in the following example, you would need to allocate a minimum of 464 bytes: 454 for overhead, 1 for the implied pen down, 8 for the coordinates, and 1 for the byte count. If you were to define an additional character(s), you would need to add the correct number of bytes for pen control, X,Y coordinates, and byte count. However, you would not need to add 454 again. Refer to the GM instruction in Chapter 3 or the ESC.T instruction in Chapter 14 for syntax details.

ManualsPlus.com

The downloadable character buffer is cleared when the plotter is initialized or set to default conditions, and upon receipt of a GM or ESC.T instruction. When the buffer is cleared, all characters in set −1 become undefined. Response to these characters in a label string follows the current fallback mode as defined in the CM instruction.

Example — Defining a Downloadable Character

The following example shows how to define the gamma symbol as a downloadable character and use it in a label string. For your reference, the character plot cell with the gamma symbol is shown after the program explanation.

```
10  'Insert configuration statement here
20  PRINT #1, CHR$(27)+".T1024;4;462:"
30  PRINT #1, CHR$(27)+".L;"
40  INPUT #1, A
50  PRINT #1, "IN;SP1;CA-1;SI.3,.4;"
60  PRINT #1, "DL65,10,0,10,30,28,30,28,24;"
70  PRINT #1, "PA1000,5000;"
80  PRINT #1, "LBThe symbol for gamma is "+CHR$(14)
              +"A"+CHR$(3)
90  PRINT #1, "SP0;"
100 END
```

NOTE: If your computer cannot read an output statement, replace lines 20 through 40 with this line: `20  PRINT #1, "GM0,500;"` ■

Program Explanation

10 defines the plotter as the system printer — change as necessary for your computer

20 allocates 1024 bytes in the I/O buffer (default), 4 bytes in the polygon buffer (the minimum), and 462 bytes in the downloadable character buffer. Note the colon (:) terminator for this instruction.

30 causes the program to pause until all of the bytes are allocated in the buffer by requesting that the plotter output the I/O buffer size when it is empty

40 reads the output response generated by line 30. (The actual response is not important, but it should be read by the computer anyway. Refer to Chapter 14 for hints on using the ESC.L instruction with the ESC.T instruction, and to Chapter 13 for methods of reading the output response.)

ManualsPlus.com

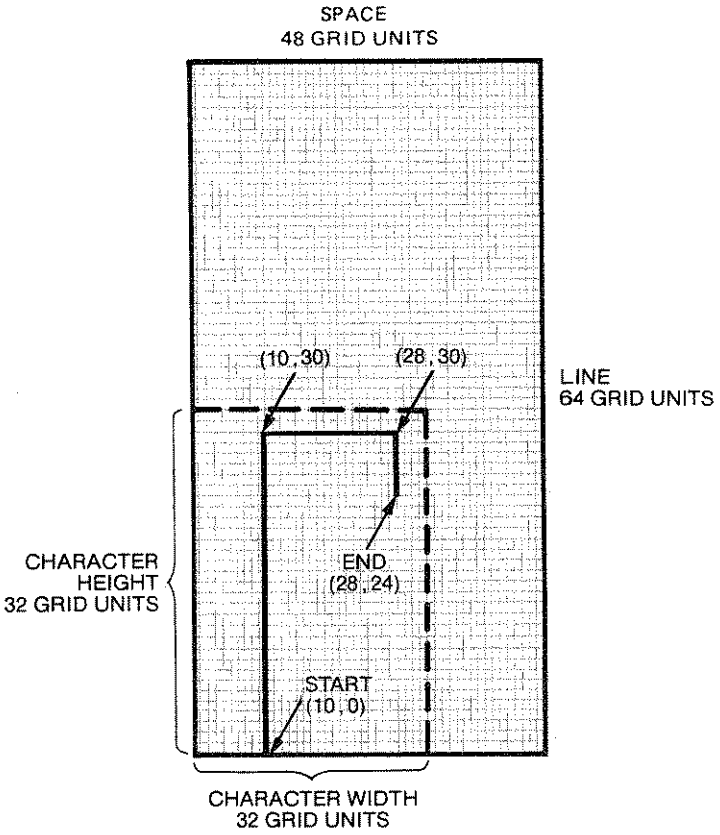
50 initializes the plotter; selects pen 1; designates the downloadable set (−1) as the alternate character set; specifies an absolute character size

60 designates decimal code 65 to be the downloadable character, then defines the gamma symbol

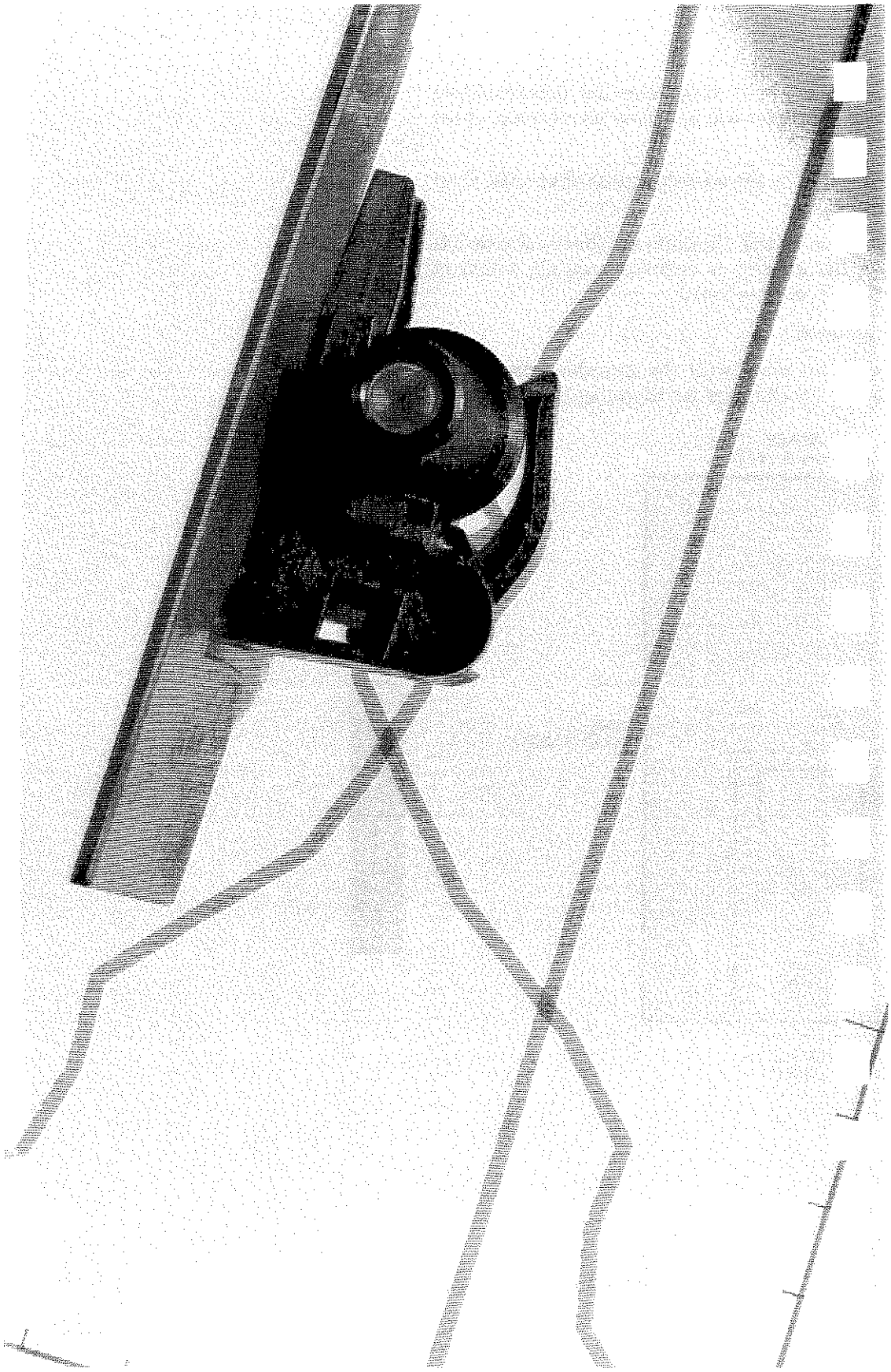
80 draws the label, shifting out to the alternate set (decimal code 14) to print the gamma — the gamma is decimal code 65, which is obtained by entering A from the keyboard

90 returns the pen to the carousel

The character plot cell is shown next, with the gamma symbol. Remember that you use absolute coordinates for defining the character.



ManualsPlus.com



ManualsPlus.com

Chapter 12
Digitizing

What You'll Learn in This Chapter

You can use your plotter as a digitizer as well as a plotter. Digitizing means moving the pen to a point on the plotting surface, entering the point, and sending the X,Y coordinates of that point to the computer.

In this chapter you'll learn how to:

- use HP-GL instructions to digitize
- verify that an X,Y coordinate point has been entered
- write a program for digitizing

The three digitize instructions are presented first, followed by programming advice.

NOTE: In order to use the plotter to digitize, your computer must be able to read data from the plotter. If you are programming in a high-level language, such as BASIC, you cannot digitize using HP Touchscreen (150) computers in an HP-IB configuration. Some computers (such as HP Series 80 computers), require an I/O ROM to obtain digitized points. ■

HP-GL Instructions Covered

DP Digitize Point Instruction
OD Output Digitized Point and Pen Status Instruction
DC Digitize Clear Instruction

Terms You Should Understand

If you are unfamiliar with any of these terms, refer to the glossary at the end of the manual.

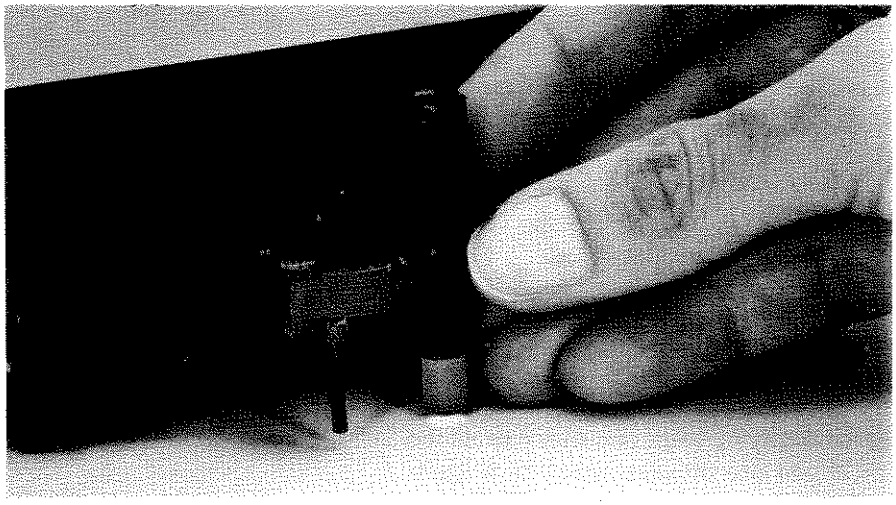
Digitize — to convert a physical position (defined by X,Y coordinates) to numerical information that can be understood by the computer.

ManualsPlus.com

Output Terminator — the character(s) sent by the plotter at the end of the response to an output instruction.

Preparing Your Plotter for Use as a Digitizer

Although you can use a pen for digitizing, we recommend you use a digitizing sight, which is available as an accessory (Part No. 09872-60066). Load the sight into the pen holder just as you would load a pen. Refer to the following picture. For highest accuracy, digitize with the sight in the pen down position.



7550-A-9-1

Loading the Digitizing Sight

To avoid smearing ink on the tip of the digitizing sight, do not load the digitizing sight directly into the pen carousel.

NOTE: If you are using the HP-IB interface, you cannot digitize while the plotter is in listen-only mode. When the plotter is in listen-only mode, it cannot send the coordinates of a digitized point to the computer. ■

The Digitize Point Instruction, DP

USES: The DP instruction allows you to digitize points on the plotter. Use this instruction to input data for a graphics program or to obtain the coordinates of a point or points on a plot.

SYNTAX: *DP term*

EXPLANATION: When the plotter receives the DP instruction, automatic pen lift and storage are suppressed, and the message **DIGITIZE**

ManualsPlus.com

POINT is displayed, along with a flashing asterisk (*) under the **ENTER** key. Use the cursor keys to move the digitizing sight to the desired position, lower the sight, and then press **ENTER**. Pressing **ENTER** sets bit position 2 of the status byte, indicating a digitized point is available for output. Use the OD instruction to *retrieve* the X,Y coordinates of the point and the pen status (up/down).

After you press **ENTER**, automatic pen lift and storage are reactivated and the digitize message is replaced with the current front-panel display.

Using a parameter with the DP instruction generates error 2 (wrong number of parameters), and the plotter executes the DP instruction anyway.

The Output Digitized Point and Pen Status Instruction, OD

USES: The OD instruction outputs the X,Y coordinates and pen status (up/down) associated with the last digitized point. Use this instruction after DP and **ENTER** to return the coordinates of the digitized point to the computer.

SYNTAX: *OD term*

RESPONSE: X,Y , pen status [TERM]

EXPLANATION: The pen position (X,Y coordinates) and status (up/down) are output to the computer as three integers in ASCII, separated by commas and followed by the output terminator. The pen position is in plotter units if the plotter is in **STANDARD** mode, and in current units if in **ENHANCED** mode. The ranges of the X,Y coordinates are the hard-clip limits of the plotter. The pen status is either 0 (pen up) or 1 (pen down).

When the plotter receives the OD instruction bit position 2 of the status byte is cleared.

After you send the OD instruction, have your program read the plotter's output response. The timing of output depends on which interface you are using (RS-232-C or HP-IB). Refer to Hints for Obtaining Plotter Output in Chapter 13, for more information.

Using parameters with the OD instruction generates error 2 (wrong number of parameters), and the plotter executes the OD instruction anyway.

ManualsPlus.com

The Digitize Clear Instruction, DC

USES: Use the DC instruction to terminate digitize mode. If you are using an interrupt routine in a digitizing program to branch to another plotting function, you could use DC to clear the digitize mode immediately after branching.

SYNTAX: *DC term*

EXPLANATION: When the plotter receives the DC instruction, digitize mode is terminated, automatic pen lift is reactivated, and the digitize message is replaced with the current front-panel display.

Using a parameter with the DC instruction generates error 2 (wrong number of parameters), and the plotter executes the DC instruction anyway.

Digitizing with the Plotter

When you are digitizing, make sure that a point has been entered before attempting to retrieve that point with the OD instruction. The three methods for doing this are explained next, and program examples are given.

Manual Method

The manual method is the easiest method of digitizing to understand. It is not efficient in applications where many points will be entered, or in cases where human intervention in program execution is not possible. To implement the manual method:

1. In a program, send a DP instruction to the plotter.
2. Next, use a statement that will cause the program to display or print a message prompting you to enter a point.
3. Follow the prompt with a statement that will cause the program to pause until instructed to continue. Using the BASIC INPUT statement and entering an empty string will work on some systems. Some versions of BASIC use statements such as STOP, WAIT, or PAUSE.
4. Move the digitizing sight (or pen) to the point to be entered, using the front-panel cursor control keys. Complete final positioning with the sight (or pen) down. Press the **ENTER** button on the plotter.
5. Now cause the program to resume. If you used an INPUT statement in step 3, press the RETURN key on the computer. The way you resume program execution depends on the statement you used to halt the program.

ManualsPlus.com

6. The next steps of the program, in order, should be an OD instruction to the plotter, a read statement by the computer to read the X,Y coordinates and the pen status, and then steps to process the digitized data.

Using this method, you do not need to monitor the status byte because the program does not proceed to the OD instruction until you enter a point and cause the program to resume. The following example uses this method.

You can also use a simpler procedure using OA or OC instead of OD. If you do, omit the DP in step 1 and pressing **ENTER** in step 4. Refer to Chapter 13 for descriptions of the OA and OC instructions.

Example — Digitizing Using the Manual Method

The following program digitizes a single point and displays the coordinates and pen status.

NOTE: Check your computer documentation to see how your computer reads input from the plotter, it may vary from the example shown here. HP Touchscreen (150) computers will require use of a separate file for output and input, only one of which may be open at a given time. Refer to Chapter 6, Plotter Interconnection, in the Operation and Interconnection Manual for an example. ■

```
10  'Insert configuration statement here
20  PRINT #1, "DP;"
30  PRINT "Enter a point, then press Return"
40  INPUT N$
50  PRINT #1, "OD;"
60  INPUT #2, X,Y,P
70  PRINT X,Y,P
80  END
```

Monitoring the Status Byte

The second method monitors bit position 2 (the third least significant bit) of the plotter's status byte, which is set when a digitized point is available. Refer to the Output Status Instruction, OS, in Chapter 13 for more information.

There are a variety of ways to monitor bit position 2, depending on the instructions available in the computer you are using. The status byte can be operated on arithmetically to check for the availability of a digitized point. Executing successive divisions of a number by a power of two and checking the answer for an odd or even integer is a common way of monitoring bits without converting the number to binary form. The following example uses this method.

ManualsPlus.com

Example — Digitizing by Monitoring the Status Byte

The following sequence of BASIC instructions will check the proper bit of the status byte. In line 50, use your computer's BASIC read statement to read the status byte into a variable called Status. (INT is a function that returns the integer portion of a number.)

```
10  'Insert configuration statement here
20  PRINT #1, "DP;"
30  PRINT "Enter a point by pressing ENTER"
40  PRINT #1, "OS;"
50  INPUT #1, STATUS
60  STATUS = INT(STATUS/4)
70  IF STATUS = INT (STATUS/2)*2 THEN 40
80  PRINT #1, "OD;"
90  INPUT #1, X,Y,P
100 PRINT X,Y,P
110 END
```

Program Explanation

10 establishes HP-IB or RS-232-C interface conditions

20 prepares plotter to accept a digitized point

30 prompts you to enter a point on the plotter and press **ENTER** on the plotter

40 sends the output status instruction

50 reads the status

60 shifts bits right by two positions

70 if a point hasn't been obtained, reads status again

80 outputs the digitized point

90 reads X,Y coordinates and pen status (up/down)

100 displays X,Y coordinates and pen status

Example — Digitizing Many Points

In many applications, a large number of points need to be digitized. When the computer is used to monitor bit position 2, the data points may or may not be processed immmediately. Generally, you need to allocate space for the total number of points to be digitized. Then, you can establish a loop to process the total number of points, calling a subroutine each time to check that a point has been entered.

A complete program follows. When prompted to enter a point, use the cursor keys to move the pen to the desired position. Now press the

ManualsPlus.com

ENTER button on the plotter. Continue for all 25 points. Their coordinates will be displayed on the computer's screen after they have all been entered.

```
10  'Insert configuration statement here
20  DIM X(25),Y(25),P(25)
30  FOR C=1 TO 25
40    PRINT #1, "DP;"
50    PRINT "Enter point ";C
60    GOSUB 140
70    PRINT #1, "OD;"
80    INPUT #1, X(C),Y(C),P(C)
90  NEXT C
100 FOR C=1 TO 25
110   PRINT X(C),Y(C),P(C)
120 NEXT C
130 END
140 ' Check bit 2 for digitized point
150 PRINT #1, "OS;"
160 INPUT #1, STATUS
170 STATUS=INT(STATUS/4)
180 IF STATUS=INT(STATUS/2)*2 THEN 150
190 RETURN
```

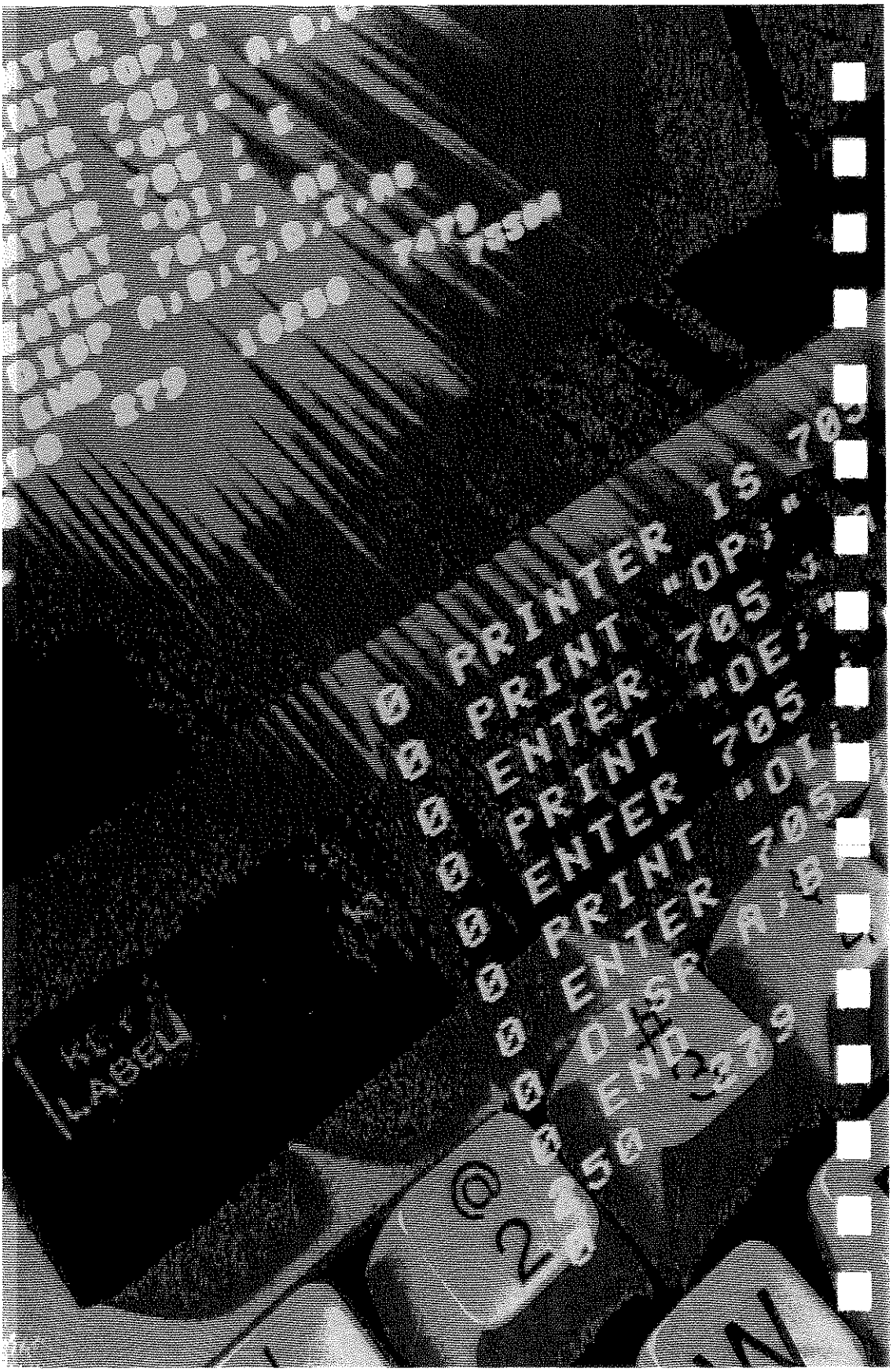
HP-IB Interrupts and Polling

Advanced programmers, who are thoroughly familiar with the HP-IB interface, polling techniques, and interrupts can use the third method, described in this section. Use this technique if your computer can perform useful tasks while waiting for the digitized point to be entered.

This method involves setting a value of 4 in the S-mask of the IM instruction (e.g. IM 223 , 4 , 0 ;) to cause the plotter to generate a service request when a digitized point is available. With an interrupt routine enabled for service requests, the computer can send a DP instruction to initiate digitizing, and then proceed with some other task until the digitized point is entered. When the point is available, the computer is interrupted by the service request, and program execution branches to the routine to process the digitized data.

This routine could simply send an OD instruction and read the digitized point, or it could perform bit checking of the plotter status byte if multiple S-mask values have been specified to generate the service request. The status byte can be obtained by serial polling or by sending an OS instruction (refer to Chapter 13). Because interrupts and polling are highly machine-dependent and beyond the scope of this manual, no examples are given. Refer also to the Input Mask Instruction, IM, in Chapter 13, and Serial and Parallel Polling in Chapter 15.

ManualsPlus.com



ManualsPlus.com

Chapter 13
Obtaining Information from the Plotter

What You'll Learn in This Chapter

In this chapter, you will learn about the conditions under which an error message, service request, and parallel poll response will occur. You will also learn how to obtain information from the plotter concerning its current pen location, error types, factors, identification, options, status, and the currently installed carousel type.

NOTE: HP Touchscreen (150) computers, when used in an HP-IB configuration, cannot read output from the plotter when programmed in a high level language. ■

HP-GL Instructions Covered

IM The Input Mask Instruction
OA The Output Actual Position and Pen Status Instruction
OC The Output Commanded Postion and Pen Status Instruction
OE The Output Error Instruction
OF The Output Factors Instruction
OI The Output Identification Instruction
OO The Output Options Instruction
OS The Output Status Instruction
OT The Output Carousel Type Instruction

Other HP-GL Output Instructions

The following output instructions are described in other chapters of this manual, depending on the type of output they produce. (For example, the output label instruction, OL, is described with the other label instructions in Chapter 7.)

OD The Output Digitized Point Instruction — Chapter 12
OG The Output Group Count Instruction — Chapter 10
OH The Output Hard-Clip Limits Instruction — Chapter 9
OK The Output Key Instruction — Chapter 10
OL The Output Label Instruction — Chapter 7

ManualsPlus.com

OP The Output P1 and P2 Instruction — Chapter 9
OW The Output Window Instruction — Chapter 9

Hints for Obtaining Plotter Output Responses

NOTE: The following paragraphs discuss the general concepts of obtaining plotter output. You should also be aware of other considerations that depend on your interface configuration (HP-IB or RS-232-C). Be sure to read the notes for each interface later in this section. ■

The HP-GL output instructions each request a certain type of information from the plotter; in other words, they request the plotter to *output* certain information. When the plotter executes an output instruction, it responds by making the requested information available. This is called the output response. If you wish to obtain the output response for use in your program, your computer must read the response with an input statement (such as ENTER, INPUT #1, READ, or READLN). Therefore, you should follow a two-step sequence when requesting information from the plotter, as follows:

1. Execute the desired output instruction just as you would execute any HP-GL instruction (for example, PRINT #1, "OE;"). Note that none of the output instructions use parameters.
2. Read the plotter's output response using an appropriate computer input statement for your language (for example, INPUT #1, A).

Check your computer documentation for the correct input statement to use, and also for any special configuration requirements.* With Microsoft® BASIC, you can use the statement INPUT #1 to read the plotter's output response.

When you read the response, you should specify the correct number and type of variables that will hold the plotter's output response. For example, if an output instruction will cause an integer to be output, you should read the response into a numeric variable. With Microsoft® BASIC, you would use a statement such as INPUT #1, A (where A is the variable that will hold the plotter's output response). Similarly, if an output instruction will cause a character string to be output, you should read the response into a string variable, such as INPUT #1, B$.

Some output instructions cause more than one piece of information to be output. If so, the plotter sends a comma between each piece of information. For example, the OC instruction causes the following output response: X-coordinate , Y-coordinate , pen status. For instructions that

*Some computers (such as the HP Series 80 computers) require an input/output (I/O) ROM in order to obtain information from the plotter.

ManualsPlus.com

cause a multiple output response, be sure to read **all** of the output response even if you only need to know part of the information. When you read the plotter's output response, it is removed (cleared) from the plotter. Thus, if you do not read all of the output response, the plotter retains any remaining portion of the response until it is obtained by a subsequent read operation or until a subsequent output response overwrites it. For the OC example, you would use a statement such as INPUT #1, A, B, C to read all three responses generated by the OC instruction.

The descriptions of each output instruction tell you what the plotter's response is. For a summary of the output responses to all HP-GL output instructions, refer to the table at the end of this chapter.

For examples of obtaining the plotter's output response, refer to The Output Label Length Instruction, OL, in Chapter 7; also refer to The Define Key Instruction, KY, and The Output Key Instruction, OK, both in Chapter 10.

Notes for HP-IB Configurations

After an output instruction is completely processed, the plotter makes the requested data available for output. However, the data is not output until the plotter is instructed to talk by a computer read statement. It is best to read the output response immediately. If the plotter receives another output instruction before the current response has been read, the current response will be overwritten with the new response generated by the new output instruction. This could cause you confusion if you send a read statement expecting a certain response, but you get a different response.

In order to send the output response to the computer when requested by a read statement, the plotter must be in an addressable mode. If the plotter is in the listen-only mode, the computer will not get a response to its read statement, and typically your program will halt. Therefore, do not send output instructions to the plotter when the plotter is in listen-only mode. (Addressable and listen-only modes are described in Chapter 15.)

The plotter signals the end of its output response with an output response terminator, denoted in this manual by [TERM]. In an HP-IB configuration, the output response terminator is a carriage return followed by a line feed (**CR LF**).

The plotter sets the EOI line high concurrently with the output of the line feed (**LF**) portion of the output terminator. This EOI signal is interpreted by some computers to mean that the read statement has been satisfied and program execution can resume. The EOI line is defined in Chapter 15. Also refer to your computer documentation for your computer's interpretation of EOI.

ManualsPlus.com

Notes for RS-232-C Configurations

As soon as an output instruction has been parsed by the plotter, output occurs according to the handshake protocol established by the ESC.P, ESC.M, or ESC.N instructions. You should use these instructions to specify turnaround delays and intercharacter delays as necessary to assure that output will not be lost because the computer is not prepared to receive it. The documentation for your computer should contain information about whether or not delays are required. Refer also to Chapter 16 of this manual for more information.

Even though output occurs automatically, you need to read the output response if you wish to use it in your program. It is best to read the output response immediately. If the plotter receives another output instruction while an original output request is being processed, error 10 will be set. In this case, the original output instruction will continue normally, but the new output instruction will be ignored.

The plotter signals the end of its output response with an output response terminator, denoted in this manual by [TERM]. In an RS-232-C configuration, the output response terminator is a carriage return (**CR**), unless altered by the ESC.M instruction (refer to Chapter 16 for an explanation of the ESC.M instruction).

The Input Mask Instruction, IM

USES: The IM instruction controls the conditions under which HP-GL error status is reported, the conditions that can cause an HP-IB service request, and the conditions that can cause a positive response to an HP-IB parallel poll. Use this instruction with either the HP-IB or RS-232-C configurations to change the conditions under which HP-GL error status is reported. In an HP-IB configuration, use the instruction to enable the plotter to send a service request when specified bits of the status byte are set, and/or enable a positive response to a parallel poll under the conditions specified. (IM is not an output instruction; i.e., it does not generate a plotter output response that must be read by the computer. However, IM does involve I/O operations, so it is included in this chapter. Refer to Chapter 15 for information on the HP-IB service request and parallel polling.)

SYNTAX: *IM* E-mask value (, S-mask value (, P-mask value)) *term*
or
IM *term*

ManualsPlus.com

Parameter	Format	Range*	Default
E-mask value	integer	0–255	223
S-mask value	integer	0–255	0
P-mask value	integer	0–255	0

*Refer to the tables on the next pages for each parameter.

EXPLANATION: When initialized or set to default conditions, the plotter automatically sets the E-mask to 223 (error numbers 1, 2, 3, 4, 5, 7, and 8 will be reported), the S-mask to 0 (none of the status-byte bits can send the service request), and the P-mask to 0 (none of the status-byte bits can cause a parallel poll response of logical 1). An IM instruction without parameters (IM;) automatically sets the values to 223,0,0.

In the RS-232-C configuration, the S- and P-masks are of no use and are ignored if present.

1. E-mask. The E-mask value specified is the sum of any combination of the bit decimal values shown below. When an HP-GL error occurs, the bit in the E-mask corresponding to the error number is tested to determine if the error bit (bit 5) of the status byte is to be set and the error is to be displayed on the front panel. Refer to The Output Status Instruction, OS, later in this chapter. If a bit is not set, there is no way to determine if that error occurred.

E-Mask Bits

Decimal Value	Bit No.	Error No.	Meaning
1	0	1	Instruction not recognized
2	1	2	Wrong number of parameters
4	2	3	Bad parameters received
8	3	4	Not used
16	4	5	Unknown character set
32	5	6	Position overflow
64	6	7	Buffer overflow
128	7	8	Not used

An E-mask value of 60 (4 + 8 + 16 + 32) specifies that errors 3 through 6 will set the error bit in the status byte and will be displayed on the front panel whenever they occur. Errors 1 and 2, however, will not set the

ManualsPlus.com

error bit or be displayed if they occur, since they are not included in the E-mask value. Errors 4 and 8 never occur. Therefore, specifying these errors in the E-mask has no effect.

2. S-mask. The S-mask value specified is the sum of any of the bit decimal values shown below. When a bit of the status byte changes value, the status byte is ANDed with the S-mask in a bit-by-bit fashion to determine if bit 6 of the status byte is to be set and the service request sent. Bit 6 is cleared by a serial poll; it will not be set again until one of the conditions specified by the S-mask value occurs again. (Refer also to The Output Status Instruction, OS, later in this chapter.)

S-Mask Bits

Decimal Value	Status Bit No.	Meaning
1	0	Pen down
2	1	P1 or P2 changed
4	2	Digitized point available
8	3	Initialized
16	4	Ready for data (buffer empty)
32	5	Error
64	6	Not used
128	7	Not used

For example, an S-mask value of 16 specifies that the ready-for-data bit (bit 4) of the status byte will set bit 6, which will send the service request message. The other bits (bits 0 through 3, bits 5 through 7) will not set bit 6.

3. P-mask. The P-mask value specifies which of the status-byte conditions will result in a logical 1 response to an HP-IB parallel poll, as shown in the next table.

ManualsPlus.com

P-Mask Bits

Decimal Value	Status Bit No.	Meaning
1	0	Pen down
2	1	P1 or P2 changed
4	2	Digitized point available
8	3	Initialized
16	4	Ready for data (buffer empty)
32	5	Error
64	6	Not used
128	7	Not used

For example, a P-mask value of 48 specifies that only bits 4 and 5 (16 + 32) of the status byte can cause the plotter to respond to a parallel poll with a logical 1 on the appropriate data line.

The following table summarizes the possible error conditions or unexpected results that you might observe with the IM instruction.

Condition	Error	Plotter Response
no parameters	none	establishes default values of 223,0,0
more than 3 parameters	2	executes first 3 parameters
parameter out-of-range	3	ignores instruction

The Output Actual Position and Pen Status Instruction, OA

USES: The OA instruction forces execution of all instructions currently in the vector buffer and then outputs the X- and Y-coordinates and pen status (up or down) associated with the actual pen position. Use this instruction to determine when the vector buffer is empty, and the pen's current position in plotter units. You might use this information to position a label or figure, to determine the parameters of some desired window, or to determine the pen's current position if it has been moved using front-panel cursor keys.

SYNTAX: *OA term*

RESPONSE: X-coordinate, Y-coordinate, pen status [TERM]

ManualsPlus.com

EXPLANATION: The pen position and status are output to the computer as integers in ASCII, separated by commas. The X- and Y-coordinates are output in plotter units; their ranges are the current hard-clip limits. The pen status is either 0 (pen up) or 1 (pen down).

If you include a parameter with the OA instruction, error 2 (wrong number of parameters) is generated, and the plotter executes the OA instruction anyway.

The Output Commanded Position and Pen Status Instruction, OC

USES: The OC instruction outputs the X- and Y-coordinates and pen status (up or down) associated with the last valid pen position instruction. Use this instruction to determine the pen's last valid commanded position in plotter units or user units, depending on whether scaling is off or on. You might use this information to position a label or figure, or determine the parameters of an instruction which moved the pen to the limits of some window. This instruction is especially useful when the pen is physically at the plotting limits and the pen position does not coincide with the instructed position, or when output in user units is desired.

SYNTAX: *OC term*

RESPONSE: X-coordinate, Y-coordinate, pen status [TERM]

EXPLANATION: The pen position and status are output to the computer as decimal numbers or integers in ASCII, separated by commas.

When scaling is off, the X- and Y-coordinates are integers in plotter units. When scaling is on, the X- and Y-coordinates are decimal numbers in user units. In either case, the range is -2^{23} to $2^{23}-1$. The pen status is either 0 (pen up) or 1 (pen down). The plotter outputs a negative sign for negative numbers; positive signs are suppressed.

If you include a parameter with the OC instruction, error 2 (wrong number of parameters) is generated, and the plotter executes the OC instruction anyway.

The Output Error Instruction, OE

USES: The OE instruction outputs the number corresponding to the first HP-GL error (if any). Use this instruction to determine the type of the first error when debugging programs.

SYNTAX: *OE term*

ManualsPlus.com

RESPONSE: error number [TERM]

EXPLANATION: When an OE instruction is received, the plotter converts the last HP-GL error to a positive integer in ASCII. The error numbers are defined below.

HP-GL Error No.	Meaning
0	No error
1	Instruction not recognized
2	Wrong number of parameters
3	Out-of-range parameter, or illegal character
4	Not used
5	Unknown character set
6	Position overflow
7	Buffer overflow
8	Not used

After execution of an OE instruction, bit position 5 of the status byte is cleared (if set), and the error message is removed from the front-panel display.

If you include a parameter with the OE instruction, error 2 (wrong number of parameters) is generated, and the plotter executes the OE instruction anyway.

The Output Factors Instruction, OF

USES: The OF instruction outputs the number of plotter units per millimetre in each axis. This instruction enables the plotter to be used with software which must know the size of a plotter unit.

SYNTAX: *OF term*

RESPONSE: 40,40 [TERM]

EXPLANATION: The plotter always outputs the ASCII integers 40 and 40, separated by a comma. These factors indicate that there are 40 plotter units per millimetre in the X-axis and in the Y-axis (0.025 mm per plotter unit).

If you include a parameter with the OF instruction, error 2 (wrong number of parameters) is generated, and the plotter executes the OF instruction anyway.

ManualsPlus.com

The Output Identification Instruction, OI

USES: The OI instruction outputs the plotter's model number. This instruction is especially useful in a remote operating configuration to determine which model of plotter is on-line.

SYNTAX: *OI term*

RESPONSE: 7550A [TERM]

EXPLANATION: The plotter always outputs its model number and letter as a 5-character string.

If you include a parameter with the OI instruction, error 2 (wrong number of parameters) is generated, and the plotter executes the OI instruction anyway.

The Output Options Instruction, OO

USES: The OO instruction outputs eight option parameters. This instruction is especially useful in a remote operating configuration to determine which options are available in the plotter that is on-line.

SYNTAX: *OO term*

RESPONSE: (Refer to the EXPLANATION below.)

EXPLANATION: The plotter always outputs the following eight ASCII integers, separated by commas:

C,1,0,0,1,1,0,1 [TERM]

- 1 — Indicates plotter has configurable graphics memory.
- 1 — Indicates plotter has polygon fill instructions.
- 0 — Indicates plotter has arc and circle instructions.
- 0 — Indicates plotter has pen select capability.
- 1 — Indicates status of plotter's paper check and paper feed bits.
- C —

C can take on a value from 0 to 3, depending on the following settings of the paper check bit and the paper feed bit. The paper check bit is set to "1" whenever the plotter has drawn a mark on a piece of paper; otherwise, it is set to "0." The paper feed bit is set to "1" when the plotter is in automatic paper feed mode; it is set to "0" when the plotter is in manual paper feed mode.

ManualsPlus.com

C	Paper Check Bit	Paper Feed Bit
0	0	0
1	0	1
2	1	0
3	1	1

For example, if you execute the OO instruction and receive a value of 2 for C, this means the plotter has drawn a mark on the paper and it is in manual paper feed mode.

If you include a parameter with the OO instruction, error 2 (wrong number of parameters) is generated, and the plotter executes the OO instruction anyway.

The Output Status Instruction, OS

USES: The OS instruction outputs the decimal equivalent value of the status byte. Use this instruction in debugging operations and in digitizing applications.

SYNTAX: *OS term*

RESPONSE: status [TERM]

EXPLANATION: Upon receipt of the OS instruction, the internal 8-bit status byte is converted to an ASCII integer between 0 and 255. The status-byte bits are defined below.

Decimal Value	Bit No.	Meaning
1	0	Pen down
2	1	P1 or P2 newly established; cleared by OP
4	2	Digitized point available; cleared by OD
8	3	Initialized; cleared by OS
16	4	Ready for data (buffer empty)
32	5	Error; cleared by OE
64	6	Request service
128	7	Not used (always set to "0")

ManualsPlus.com

When the plotter is first turned on, the status is 26: the sum of 2 (P1 and/or P2 newly established), 8 (initialized), and 16 (ready for data).

Upon execution of an OS instruction by the plotter, bit position 3 is cleared and the status is 18: the sum of 2 (P1 and/or P2 newly established) and 16 (ready for data).

If you include a parameter with the OS instruction, error 2 (wrong number of parameters) is generated, and the plotter executes the OS instruction anyway.

The Output Carousel Type Instruction, OT

USES: The OT instruction outputs the current carousel type and stall occupancy information. This instruction is especially useful in a remote operating configuration to determine the carousel type and occupied pen stalls.

SYNTAX: *OT term*

RESPONSE: type, map [TERM]

EXPLANATION: The current carousel type and its pen map are output as two ASCII integers, separated by commas.

The type field can contain the values −1 through 4, which have the following meanings:

Type Value	Meaning
−1	Carousel of unknown type is installed
0	Carousel is not installed
1	Carousel for paper fiber-tip pens is installed
2	Carousel for roller-ball pens is installed
3	Carousel for drafting pens is installed
4	Carousel for transparency fiber-tip pens is installed

The map value is defined as the sum (0 through 255) of any combination of the bit decimal values shown on the next page. For example, a map value of 15 (1 + 2 + 4 + 8) indicates pens are installed only in stalls 1, 2, 3, and 4 of the carousel. A map value of 0 means either all stalls are empty, or no carousel is installed.

ManualsPlus.com

Decimal Value If Occupied	Pen Stall Number
1	1
2	2
4	3
8	4
16	5
32	6
64	7
128	8

If you include a parameter with the OT instruction, error 2 (wrong number of parameters) is generated, and the plotter executes the OT instruction anyway.

Summary of Output Response Types

The following table shows the number and type of items in the response to each HP-GL output instruction. The table includes all output instructions explained in Chapters 7, 9, 10, 12, and 13. This table will be helpful when programming in languages (such as FORTRAN) that require you to specify the type and number of digits in a variable.

Instruction	Number of Parameters Returned*	Type and Range
OA	3	integers: 1st and 2nd, ⩽5 digits; 3rd, 1 digit
OC	3	1st and 2nd: decimal, integer portion ⩽8 digits and fractional portion ⩽4 digits. 3rd: integer, 1 digit
OD	3	integers: 1st and 2nd, ⩽5 digits; 3rd, 1 digit
OE	1	integer: 1 digit
OF	2	integers: 2 digits each

*In addition to these parameters, the output terminator [TERM] is always sent at the end of output, and commas are sent to separate parameters.

(Table continued)

ManualsPlus.com

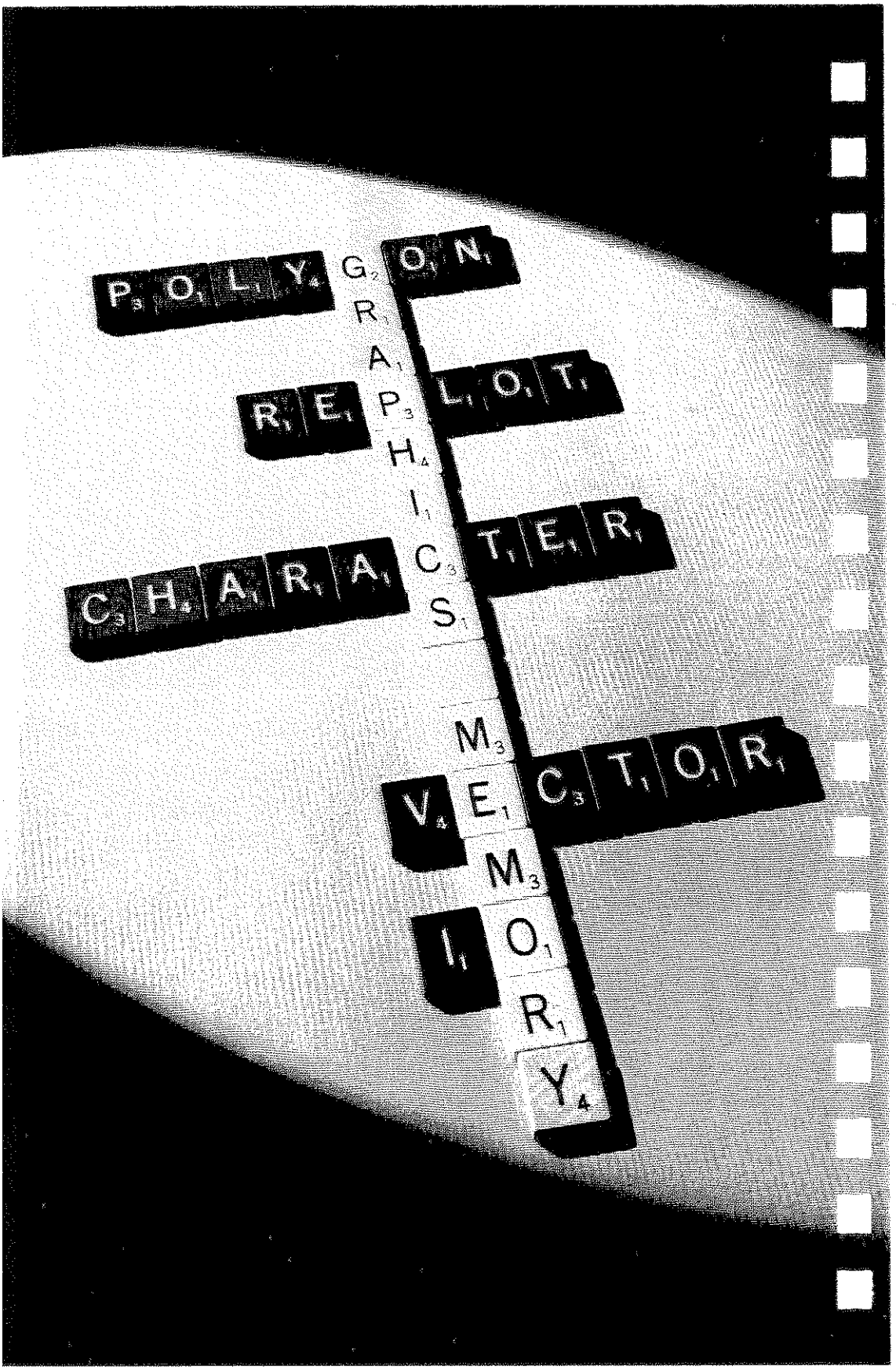
Instruction	Number of Parameters Returned*	Type and Range
OG	2	integers: 1st, ⩽ 8 digits; 2nd, 1 digit
OH	4	integers: all ⩽ 5 digits
OI	1	5-character string
OK	1	integer: 1 digit
OL	3	1st: decimal, integer portion ⩽ 3 digits and fractional portion ⩽ 4 digits. 2nd and 3rd: integers ⩽ 3 digits
OO	8	integers: 1 digit each
OP	4	integers: all ⩽ 8 digits
OS	1	integer: ⩽ 3 digits
OT	2	integers: 1st, 1 digit; 2nd, ⩽ 3 digits
OW	4	integers: all ⩽ 8 digits

*In addition to these parameters, the output terminator [TERM] is always sent at the end of output, and commas are sent to separate parameters.

ManualsPlus.com

Notes

ManualsPlus.com



ManualsPlus.com

Chapter 14
Device-Control Instructions

What You'll Learn in This Chapter

This chapter introduces you to device-control instructions. These instructions differ from HP-GL in that they control internal plotter conditions, such as: the configurable graphics memory, input/output, and data flow. HP-GL, on the other hand, generally controls functions that have a more obvious effect on plotting, such as: pen movement, scaling, labels, arcs, circles, and lines. In this chapter, you will learn about the purpose of device-control instructions and how to use these instructions to accomplish various tasks. For example, with device-control instructions you can change the size allocations of the five buffers in the configurable graphics memory, you can obtain information about the plotter's I/O errors and extended status, and you can establish monitor mode (a program debugging tool).

NOTE: This chapter presents the device-control instructions that are useful in *either* interface configuration (HP-IB or RS-232-C). Unless otherwise noted, these instructions are interpreted and operate in the same manner in both interface configurations. If you are using the RS-232-C configuration, you should refer to Chapter 16 for additional device-control instructions that are useful in RS-232-C interfacing. ■

Device-Control Instructions Covered

ESC.T The Allocate Configurable Memory Instruction
ESC.L The Output Buffer Size When Empty Instruction
ESC.S The Output Configurable Memory Size Instruction
ESC.@ The Set Plotter Configuration Instruction
ESC.B The Output Buffer Space Instruction
ESC.E The Output Extended Error Instruction
ESC.O The Output Extended Status Instruction
ESC.Q The Set Monitor Mode Instruction
ESC.K The Abort Graphics Instruction
ESC.R The Reset Instruction
ESC.A The Output Identification Instruction
ESC.U The End Flush Mode Instruction

ManualsPlus.com

Terms You Should Understand

Parse — to "decode" an instruction and its parameters in a sequential manner.

Execute — to complete the parsing of an instruction and cause the requested operation to be performed.

Buffer — an area in the plotter's memory that is reserved for data storage. This plotter has five unique buffers, described in this chapter under The Allocate Configurable Memory Instruction, ESC.T.

I/O Buffer — the buffer where all incoming graphics (HP-GL) instructions are stored, parsed, and executed on a first-in/first-out basis. There are two concepts associated with the I/O buffer: *physical* and *logical*. The physical I/O buffer is the actual, physical buffer area in the plotter's memory. The logical I/O buffer is the portion of the physical I/O buffer that limits the number of HP-GL instructions that can be stored while waiting to be parsed (HP-IB and RS-232-C configurations), and that determines when to handshake (RS-232-C configurations). Usually the physical and logical I/O buffers are identical. In most applications, this is desirable. However, it is possible to set a smaller size for the logical I/O buffer. Note that the physical and logical I/O buffers are *not* separate buffers in the plotter's memory. Instead, you should think of the logical I/O buffer as a "subset" of the physical I/O buffer. For this reason, the logical I/O buffer can never be larger than the physical I/O buffer. You can control the size of the physical I/O buffer with the ESC.T instruction, and the logical I/O buffer with the ESC.@ instruction.

What Is a Device-Control Instruction?

Device-control instructions serve two basic purposes: to control memory (buffer) allocation, and to control data transfer between the computer and the plotter (handshaking operations). In addition, some device-control instructions provide information about certain internal plotter conditions (similar to the HP-GL output instructions).

Device-control instructions are executed immediately upon receipt by the plotter (as opposed to entering the I/O buffer and being executed on a first-in/first-out basis, as is the case with HP-GL instructions). The plotter recognizes these instructions because they begin with the single ASCII escape character, **ESC** (decimal code 27). Following the escape character, each instruction includes a period (.) and a capital letter or other character. Some instructions also include parameters and a terminator. The device-control instructions follow specific syntax conventions, which are different from the syntax followed by HP-GL. Device-control syntax is presented later in this chapter, before the individual descriptions of each instruction.

ManualsPlus.com

How and When Do You Send a Device-Control Instruction to the Plotter?

The principles for sending device-control instructions to the plotter are the same as for HP-GL: you must send them as a literal string in a computer output statement such as PRINT, PRINT #, OUTPUT, WRITE, or WRITELN (discussed in Chapter 1). Accessing the character **ESC** is similar to accessing any other single ASCII character. You have two options: use a computer language-dependent function such as CHR$(27) to send the decimal code, or use your keyboard to send the character. On some computers, you can press a single escape key. On others, you can access **ESC** by pressing a combination of keys; for example, on HP Series 200 computers, you can press **CTRL** [. For more information on these options, refer to How to Send the Label Terminator and Other Nonprinting Characters under the LB instruction in Chapter 7.

You should send any device-control instructions that set plotter configurations to the plotter *before* you send any HP-GL instructions. Examples of device-control instructions that set plotter configurations are:

- Handshaking protocol instructions* — ESC.@, ESC.H, ESC.I, ESC.M, ESC.N, ESC.P, ESC.(, ESC.Y, ESC.), and ESC.Z
- Buffer allocation instructions — ESC.T, ESC.@, and ESC.R
- Monitor mode instructions — ESC.Q and ESC.@

Other instructions, such as those that request the plotter to output certain information, can be sent to the plotter at any time.

How Do You Receive Information from the Plotter?

All device-control instructions with the word "output" in their title will cause information to be output. Again, the principles for receiving this information from the plotter are the same as for HP-GL: to use the plotter's output response in a program, you must enter it into the computer with a computer input statement such as ENTER, INPUT #, READ, or READLN. Note that an output response caused by a device-control instruction is available immediately. If you do not read the information immediately, it could get replaced by a subsequent output response, or cause an error to occur.

Refer to Hints for Obtaining Plotter Output Responses in Chapter 13 for more complete information and for considerations to be aware of depending on the interface configuration you are using (HP-IB or RS-232-C).

*All of these instructions except ESC.@ are presented in Chapter 16. ESC.@ is presented in this chapter.

ManualsPlus.com

Syntax for Device-Control Instructions

A complete device-control instruction includes at least the three-character sequence consisting of "**ESC**" and "." followed by the capital letter or character indicating its function (e.g., S, T, or @). Some instructions also include parameters and a terminator, according to the syntax conventions listed below.

These syntax conventions are used with the device-control instructions in this chapter and in Chapter 16.

ESC — Denotes the single ASCII character, escape (decimal code 27). On many computers you can access this character by striking a single key on the keyboard, or you can use a function such as CHR$ to send its decimal code.

[ ] — Brackets indicate that all enclosed parameters are optional.

( ) — Parentheses indicate that each individual parameter is optional.

; — The semicolon delimits parameters. If a semicolon appears without a parameter, the parameter is defaulted.

NOTE: There is no semicolon between the three-character command sequence (e.g., **ESC** . M) and the first parameter. ■

: — The colon terminates any instruction which may have parameters and can occur after any valid number of parameter entries. Any parameter that is not specified is defaulted. (Some instructions never have parameters. The colon terminator is not used with these instructions.)

<DEC> — This symbol specifies a decimal value parameter. For example, the characters 10 would represent the decimal value ten; the characters 13 would represent the decimal value thirteen.

<ASC> — This symbol specifies the decimal code for an ASCII character (see the ASCII code definitions in Appendix A). In this case, the characters 10 would represent the ASCII line-feed character, **LF**, and 13 would represent the ASCII carriage-return character, **CR**.

ManualsPlus.com

...	Specifies any number of optional parameters. Each parameter must be followed by a delimiter (;) or the terminator (:).
[TERM]	All HP-IB output responses include a carriage return and line feed [CR LF] as a terminator. Unless changed by an ESC.M instruction, all RS-232-C output responses include a carriage return [CR] as a terminator.
Default Values; Omitting Parameters	Any parameter may be omitted by entering only the semicolon delimiter; this causes the parameter to be set to its default value. (Thus, unlike HP-GL, it *is* possible to send an instruction such as ESC.T;;2000;4000:. The parameters that have been omitted are set to their default values.) All parameters may be omitted and therefore set to default values by entering only the colon terminator after the instruction.

The Allocate Configurable Memory Instruction, ESC.T

USES: The ESC.T instruction allocates memory in the five separate buffers of the configurable graphics memory. Use this instruction to customize memory allocation to fit the requirements of your application (e.g., to increase the replot buffer and decrease all other buffers).

SYNTAX: ESC.T [(<DEC>);(<DEC>);(<DEC>);(<DEC>);(<DEC>)]:

Parameter*	Range	Default
physical I/O buffer	2–12752	1024
polygon buffer	4–12754	1778
downloadable character buffer	0–12750**	0
replot buffer	0–12750	9954
vector buffer	44–12794	44

*These parameters are listed in the required order for the ESC.T instruction.

**The minimum is 0 when there are no downloadable character definitions (when no DL instruction will be executed). However, if there are one or more downloadable character definitions, the minimum is 454.

ManualsPlus.com

EXPLANATION: The plotter's configurable graphics memory provides a total of 12800 bytes that can be divided between five buffers. You can vary the number of bytes in each buffer, depending on the requirements of your application. However, the total number of bytes for all five buffers cannot exceed 12800. If this happens, the plotter uses an algorithm to decrease the total number of bytes to a maximum of 12800. The algorithm is described at the end of this section.

As shown in the table on the previous page, each buffer requires a minimum number of bytes. If you specify fewer bytes than the minimum allowed, the plotter allocates the minimum. Also, if you specify an odd number of bytes for a buffer, the plotter allocates the next lower even number (for example, if you specify 145 for the polygon buffer, the plotter allocates 144 bytes).

The individual buffers are described next. Following these descriptions, you will find some hints and considerations that you should be aware of when executing the ESC.T instruction.

1. Physical I/O Buffer. The first parameter specifies the number of bytes to be allocated in the physical I/O buffer. The default size of the physical I/O buffer is 1024 bytes. This is a convenient size for both interface configurations. Setting the physical I/O buffer too small slows down data transmission. If you are not using any of the other buffers, consider setting the physical I/O buffer larger. The larger the buffer, the more data the computer can send to the plotter at a given time. This means the computer could send all of its data to the plotter more quickly, freeing the computer to perform other tasks while the plotter is parsing and executing the data.

   NOTE: You should be aware of the relationships between the *physical* and *logical* I/O buffers. The *physical* I/O buffer is the actual portion of the configurable graphics memory where storage and parsing of HP-GL instructions takes place. There is also a *logical* I/O buffer, which can be thought of as the operational subset of the physical I/O buffer (it is *not* a separate buffer). The size of the logical I/O buffer (*not* the size of the physical I/O buffer) is what limits plotter I/O functions such as the number of HP-GL instructions waiting to be parsed, and threshold levels in certain handshaking methods.

   The logical I/O buffer cannot be larger than the physical I/O buffer. Therefore, if you decrease the size of the physical I/O buffer, the logical I/O buffer automatically decreases to the same size. However, if you increase the size of the physical I/O buffer, the logical I/O buffer does not change unless you execute the ESC.@ instruction. Since the logical I/O buffer size is the limiting factor, you should always increase it with ESC.@ whenever you increase the physical I/O buffer with ESC.T. Refer to The Set Plotter Configuration Instruction, ESC.@, later in this chapter. ■

ManualsPlus.com

2. Polygon Buffer. The second parameter specifies the number of bytes to be allocated in the polygon buffer. All polygons defined by the wedge, rectangle, and polygon instructions (WG, EW, RA, RR, EA, ER, and PM) are stored in the polygon buffer. If your program *does not* use any of these instructions, reduce the size of this buffer so that other buffers can utilize the unused bytes. If your program *does* use any of these instructions, allocate enough bytes for the largest polygon. A good approximation is to allocate 14 bytes for every point in the largest polygon. However, a more accurate formula is provided in the last section of Chapter 6, titled The Polygon Buffer. Refer to that section, also, to learn how to determine the number of points in your largest polygon. The default size of the polygon buffer is 1778 bytes, which is sufficient for a polygon with 127 points.

3. Downloadable Character Buffer. The third parameter specifies the number of bytes to be allocated in the downloadable character buffer. The default size is 0 bytes. Therefore, if your program defines downloadable characters with the DL instruction, you must allocate enough bytes to store all of the downloadable characters that are defined. The number of bytes depends on a number of factors, but must be at least 454. A formula is provided in the section titled Allocating Memory for Downloadable Characters under the DL instruction in Chapter 11.

4. Replot Buffer. The fourth parameter specifies the number of bytes to be allocated in the replot buffer. The default size is 9954 bytes. However, after the plotter allocates the specified number of bytes to the other four buffers, it automatically allocates any excess bytes to this buffer to ensure that this buffer is as large as possible. If you do not intend to replot a drawing, you can reduce this buffer to 0 bytes. Be aware that unusually long plots could be too large to fit in even the largest possible replot buffer. (You can buffer plots and then replot them using the BF and RP instructions, or using front-panel function keys.)

5. Vector Buffer. The fifth parameter specifies the number of bytes to be allocated in the vector buffer. The endpoints of the vectors to be plotted are calculated in the vector buffer. (You can think of a vector as a line segment. Everything that is plotted, even circles and characters, is actually composed of a series of vectors, whose endpoints the plotter calculates internally.) Although the default size of the vector buffer is always 44 bytes, the capacity of the buffer in terms of the number of vectors it can hold depends on whether the curved line generator is off (normal plotting) or on. The differences between plotting with the curved line generator off and on are described next, followed by hints about the size of the vector buffer.

   In normal plotting, the pen stops at the endpoints of each vector as they become available from the vector buffer. Because of this, you can

ManualsPlus.com

sometimes see a difference in line density at the endpoints, particularly with curved lines such as arcs and circles. This effect can be minimized by using the curved line generator, which is enabled with the CV instruction (described in Chapter 10). The curved line generator causes vectors to be gathered before being released for plotting as a group. The pen does not stop at each endpoint in this group of vectors, so the curve appears smoother.

When the curved line generator is off, the plotter requires 22 bytes per vector. Thus, the default vector buffer size of 44 bytes will hold 2 vectors. This is sufficient for most applications.

When the curved line generator is on, the plotter requires 10 bytes per vector. Thus, the default vector buffer size will hold 4 vectors. If you turn on the curved line generator with the CV instruction, you should increase the vector buffer so that it will hold the largest group of vectors that you are plotting; if you do not increase the vector buffer, the benefit of turning on the curved line generator will be negligible. (For example, if you are plotting a circle with a 5-degree chord tolerance, you should increase the vector buffer to 720 bytes in order to hold the 72 vectors that define the circle.)

Note that certain front-panel functions (such as pen select, pen speed, P1, and P2) are added to the vector buffer and are not executed until the current vectors have been released. Therefore, when the size of the vector buffer is large and it becomes full, the plotter will not respond immediately when you press a front-panel function key.

Hints for Using the ESC.T Instruction to Allocate Memory

When the ESC.T instruction is executed, it performs the following operations in this sequence:

- Clears the I/O buffer.
- Waits for the vector buffer to empty.
- Clears each buffer in the configurable graphics memory, and allocates memory as specified in its parameters.
- Resets the parser.

Because of this sequence of operations, there are two programming hints that you should be aware of, described next.

1. **Allocate buffer sizes at the beginning of a program, when the buffers are empty — *before* you send any HP-GL instructions to the plotter.** There are two reasons for this. First, if the buffers are not empty, all data stored in them will be lost when they are cleared by ESC.T. Second, if you execute an instruction that makes use of

ManualsPlus.com

buffer space and you have not previously allocated enough space, you will receive error 7 (buffer overflow). This error usually clears the buffer, which means that instructions stored in that buffer (or requiring use of that buffer) will not be executed. If you receive error 7, check the buffer allocations. You can obtain the current allocations with the output configurable memory size instruction, ESC.S, described later in this chapter.

2. **After executing the ESC.T instruction, execute the ESC.L instruction.** The ESC.T instruction can take a few milliseconds to clear the buffers and complete all memory allocations. Therefore, your program should wait for this to happen before sending any data to the plotter. The easiest way to accomplish this is with the ESC.L instruction. This instruction waits until the I/O buffer is empty and then outputs the size of the logical I/O buffer. Therefore, the ESC.L instruction does not provide an output response until the ESC.T instruction has finished. In this application of the ESC.L instruction, the numerical value of the logical I/O buffer size is not important. However, your program should read the output response before sending more data, in order to avoid potential errors. For examples of allocating memory with ESC.T and ESC.L, refer to The Fill Polygon Instruction, FP, in Chapter 6, and to The Define Downloadable Character Instruction, DL, in Chapter 11. The ESC.L instruction is described next in this chapter.

   Note that you can also execute an ESC.L instruction *before* an ESC.T instruction. You would do this if you suspect the buffers might not be empty and want to prevent the data loss that would occur when ESC.T is executed. For example, if you send your program to the plotter while the plotter is completing a previous plot, the ESC.T instruction in your program could cause data for the previous plot to be lost, unless you execute ESC.L before ESC.T.

   Instead of executing ESC.L, you can also determine when the I/O buffer is empty by checking bit 3 of the extended status word. You would do this with the ESC.O instruction. However, this method could be more inefficient, because you must poll the plotter continually until the response to the ESC.O instruction indicates that bit 3 has been set (i.e., that the I/O buffer is empty and ready for data). If you use this method, be sure to cause the program to wait about 0.1 seconds between successive ESC.O instructions; this allows the plotter time to execute other instructions. In BASIC, you can use a WAIT statement or read an empty string, INPUT #1, N$. If you did not insert a wait, the plotter could spend most of its time responding to the ESC.O instruction and the buffer would take a long time to become empty.

ManualsPlus.com

How the Plotter Allocates Memory When More Than 12 800 Bytes Have Been Specified

When you specify the number of bytes for each buffer, you should remember that the total number of bytes available for all five buffers is 12 800. If you execute an ESC . T or a GM instruction and the sum of the five buffer is greater than 12 800, the plotter uses the algorithm described next to reduce the sum to 12 800.

First, the plotter determines the number of excess bytes. Then, it reduces the largest parameter by the excess number of bytes. Note that this could result in this buffer being reduced to the minimum possible, which might create a buffer overflow error when you try to run your program. If there is still an excess, the plotter reduces the next largest parameter by the excess, and so on until the sum for all five parameters is equal to 12 800. The following example illustrates this algorithm:

If you send:	**ESC** . T 7024 ; 7050 ; 7076 ; 0 ; 100 :
The excess number of bytes is:	(7024 + 7050 + 7076 + 100) − 12 800 = 21 248 − 12 800 = 8452
The plotter reduces the largest parameter (the 3rd parameter, the downloadable character buffer) to its minimum; the result is:	**ESC** . T 7024 ; 7050 ; 0 ; 0 ; 100 :
Now the excess is:	(7024 + 7050 + 100) − 12 800 = 14 174 − 12 800 = 1374
The plotter reduces the next largest parameter (the 2nd parameter, the polygon buffer) by the excess (7050 − 1374 = 5674), so the final buffer allocations are:	**ESC** . T 7024 ; 5676 ; 0 ; 0 ; 100 :

Compare this result with the original parameters. If you now try to define a downloadable character with the DL instruction, the plotter will return an error 7 (buffer overflow). You might also generate an error 7 if you execute an instruction that uses the polygon buffer.

ManualsPlus.com

The Output Buffer Size When Empty Instruction, ESC . L

USES: The ESC . L instruction outputs the size, in bytes, of the logical I/O buffer when it is empty. Use this instruction to determine when the I/O buffer is empty. This is particularly helpful to determine when a memory allocation (ESC . T or ESC . R) or a page feed (PG, AF, or AH) instruction has been completed before allowing the plotter to parse the next instruction.

SYNTAX: **ESC** . L

RESPONSE: <DEC> [TERM]

EXPLANATION: The ESC . L instruction outputs an integer which is the size of the logical I/O buffer. The response is not transmitted by the plotter until the buffer is empty.

When the plotter is turned on, the output response is 1024. However, the response can be as low as 2 if either the ESC . T or ESC . @ instruction has reduced the size of the I/O buffer, or as large as 12752 if both the ESC . T and ESC . @ instructions have increased the size of the I/O buffer.

The Output Configurable Memory Size Instruction, ESC . S

USES: The ESC . S instruction outputs the total number of bytes allocated in the configurable graphics memory, or the number of bytes in any of its five buffers. Use this instruction to determine current memory allocation or to confirm the allocation performed by the GM, ESC . T, or ESC . R instructions.

SYNTAX: **ESC** . S (<DEC>):

Parameter	Range	Default
buffer size	0–5	0

RESPONSE: <DEC> [TERM]

EXPLANATION: The ESC . S instruction outputs the number of bytes allocated in the buffer specified by its parameter. This response is an integer with a value between 0 and 12800.

Executing the ESC . S instruction without any parameters (**ESC** . S:) causes the plotter to output the total number of bytes in all five buffers of the configurable graphics memory. The valid range of parameters is listed on the next page.

ManualsPlus.com

0 requests total number of bytes in all five buffers of the configurable graphics memory

1 requests number of bytes in the physical I/O buffer

2 requests number of bytes in the polygon buffer

3 requests number of bytes in the downloadable character buffer

4 requests number of bytes in the replot buffer

5 requests number of bytes in the vector buffer

Parameters less than 0 generate error 12 (invalid byte in I/O control) and cause the value 12800 to be output. Parameters greater than 5 cause the value 0 to be output.

The Set Plotter Configuration Instruction, ESC . @

USES: In an HP-IB configuration, the ESC.@ instruction sets the logical I/O buffer size and controls the parse and receive monitor modes. Use this instruction to change the logical I/O buffer size (particularly if you have increased the physical I/O buffer size with the ESC.T instruction). You can also use this instruction to enable or disable a monitor mode (however, it is easier to use the set monitor mode instruction, ESC.Q, described later in this chapter).

In an RS-232-C configuration, the ESC.@ instruction does all of the above, plus it controls hardwire handshaking and block I/O error checking. Use this instruction as described above, to change hardwire handshake conditions, and to enable or disable block I/O error checking.

SYNTAX: ESC.@ [(<DEC>);(<DEC>)]:

Parameter	Range	Default
logical I/O buffer	2–12752	1024 (or size of current physical I/O buffer)
I/O conditions	0–127	depends on operating state of plotter

EXPLANATION: The two parameters of the ESC.@ instruction are interpreted as follows.

1. Logical I/O Buffer. This parameter specifies the size of the logical I/O buffer, which is the operational subset of the physical I/O buffer. (The differences between the two buffers are described under The Allocate Configurable Memory Instruction, ESC.T, presented earlier in this chapter.) If you do not specify this parameter, the logical I/O buffer will be set to the same size as the current physical I/O buffer. When

ManualsPlus.com

the plotter is turned on, this size is 1024 bytes. In most cases, you should specify the physical and logical I/O buffers to be the same size. If you wish to increase the size of the logical I/O buffer, you should *first* increase the size of the physical I/O buffer, because the logical I/O buffer cannot be larger than the physical I/O buffer. If the physical I/O buffer has been decreased with the ESC.T instruction, the logical I/O buffer automatically decreases to the same size.

If you specify a parameter of 0 or less, error 12 (invalid byte in I/O control) is generated and the logical I/O buffer size is set equal to the current physical I/O buffer size. If you specify a parameter of 1, the logical I/O buffer size is set to 2 (the minimum possible).

For most applications, you should be sure the physical and logical I/O buffers are both set to the same size. However, if you are using the RS-232-C configuration and an Xon-Xoff handshake, you could set the logical I/O buffer size slightly smaller than the physical I/O buffer size. This would be necessary only in a situation where many characters are being transmitted after the Xoff trigger character. These characters might normally be lost when the logical and physical I/O buffer sizes are equal. However, since the threshold level is determined from the *logical* I/O buffer size, if the logical I/O buffer were slightly smaller than the physical I/O buffer, the extra characters would not get lost (because they would still be contained in the "extra" area provided by the physical I/O buffer). Refer to Chapter 16 for information on the Xon-Xoff handshake.

2. I/O Conditions. This parameter specifies a decimal equivalent value that controls the states of bits 0 through 4 of the configuration byte. These bits control hardwire handshaking, two mutually exclusive monitor modes, and block I/O error checking, as shown in the table on the next page.

   If you omit this parameter, the plotter sets defaults according to the current operating state of the plotter. The current operating state depends on the conditions selected on the front panel or established by device-control instructions such as the most recent ESC.P and ESC.Q. (When the plotter is first turned on, the default states of bits 0 and 1 are determined by the settings of the front-panel **HANDSHAKE** and **MODEM/DIRECT** function keys, whereas bits 2, 3, and 4 are set to the default state "0.")

   NOTE: In an HP-IB configuration, only bits 2 and 3 are recognized; bits 0, 1, and 4 are ignored. In an RS-232-C configuration, all bits are recognized. If you wish only to enable a monitor mode (bits 2 and 3), it is more efficient to use the set monitor mode instruction, ESC.Q, described later in this chapter. ■

ManualsPlus.com

Bit No.	Logic State	Decimal Value	Description
*0	0	0	(*RS-232-C only*). Disable hardwire handshake. (Plotter will ignore DTR line, pin 20.) Default when any handshake except hardwire has been established.
	1	1	(*RS-232-C only*). Enable hardwire handshake. (Plotter can hold off data from the computer by using the DTR line, pin 20.) Default when hardwire handshake has been established.
*1	0	0	(*RS-232-C only*). Computer or modem can hold off data from the plotter by using the CTS and DSR lines (pins 5 and 6). Default when modem has been established.
	1	2	(*RS-232-C only*). Computer or modem cannot hold off data from the plotter using the CTS and DSR lines (pins 5 and 6). Default when direct mode has been established.
**2	0	0	Specify parse monitor mode (all data is displayed as it is parsed).
	1	4	Specify receive monitor mode (all data is displayed as it is received).
**3	0	0	Disable monitor mode.
	1	8	Enable the monitor mode specified by bit 2.
***4	0	0	(*RS-232-C only*). Disable block I/O error checking.
	1	16	(*RS-232-C only*). Enable block I/O error checking.

*Rather than using ESC . @, you can probably use the front panel or other device-control instructions to establish the conditions controlled by bits 0 and 1. Refer to Chapter 16 for more information.

If either **PARSE MODE or **RECEIVE MODE** is selected on the front panel, you cannot use the ESC . @ or ESC . Q instructions to disable monitor mode, nor to change the type of monitor mode. For descriptions of the monitor modes, refer to The Set Monitor Mode Instruction, ESC . Q, later in this chapter.

***For more information, refer to The Output Extended Error Instruction, ESC . E, later in this chapter.

ManualsPlus.com

Example — Increasing the Size of the Logical I/O Buffer and Setting the Configuration Bits

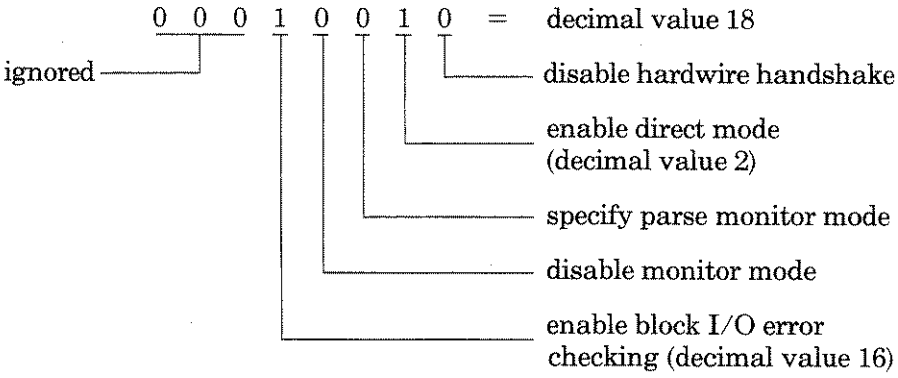
Whenever you wish to increase the size of the logical I/O buffer, you must first increase the size of the physical I/O buffer. In the following sequence of instructions, ESC . T increases the size of the physical I/O buffer to 5000 bytes (and defaults the other buffers), ESC . L ensures the allocation is complete, and ESC . @ increases the size of the logical I/O buffer to 5000 bytes (and defaults the configuration bits). Remember to add a computer-dependent read statement after ESC . L to read the output response, although the actual response is not important here.

```
ESC . T 5000 : ESC . L ESC . @ 5000 :
```

If you are using the RS-232-C interface, you might also use the ESC . @ instruction to set some of the configuration bits. The following sequence of instructions increases the logical I/O buffer size to 5000 *and* enables both direct mode and block I/O error checking. (If you are using the HP-IB interface, the only bits you can set are bits 2 and 3, the monitor mode bits; however, it is easier to use the ESC . Q instruction than the ESC . @ instruction when you wish to enable or disable a monitor mode.)

```
ESC . T 5000 : ESC . L ESC . @ 5000 ; 18 :
```

The decimal value 18 specifies the corresponding binary value to set the logic states of bits 0 through 3 as follows:



The Output Buffer Space Instruction, ESC . B

USES: The ESC . B instruction outputs the currently available space in the logical I/O buffer without waiting for the buffer to become empty. Use this instruction to confirm the size of the logical I/O buffer when you have changed the size with the ESC . @ instruction, or when the logical I/O buffer may have been automatically reduced by a reduction

ManualsPlus.com

in the physical I/O buffer (with the ESC.T instruction). You can also use this instruction in a software checking handshake to interrogate the plotter regarding available logical I/O buffer space.

SYNTAX: **ESC**.B

RESPONSE: <DEC> [TERM]

EXPLANATION: The ESC.B instruction outputs an integer which is the number of unused bytes remaining in the logical I/O buffer. When the plotter is turned on, the output response is 1024. However, the response can be as low as 2 if either the ESC.T or the ESC.@ instruction has reduced the I/O buffer, or as large as 12 752 if both the ESC.T and ESC.@ instructions have increased the I/O buffer.

If your application requires that your program continually interrogate the plotter until the response to ESC.B indicates a specific amount of available space, you should cause the program to wait about 0.1 seconds between successive ESC.B instructions. (In BASIC, use the WAIT statement or read an empty string, INPUT #1, N$.) The wait allows the plotter time to execute other instructions, thus changing the amount of available space.

The Output Extended Error Instruction, ESC.E

USES: The ESC.E instruction outputs a number that defines any I/O error related to device-control instructions. Use this instruction in program debugging to determine the errors that occur. In addition, if you are using the RS-232-C configuration, you can use this instruction in conjunction with the ESC.@ instruction to perform block I/O error checking.

SYNTAX: **ESC**.E

RESPONSE: <DEC> [TERM]

EXPLANATION: The ESC.E instruction outputs the I/O error and clears the error message (if any) from the display. (ESC.E does not affect HP-GL error conditions; they must be cleared by the OE instruction.) The I/O error is output as an integer. The range is 0 or 10–18, as described in the table on the following page.

ManualsPlus.com

Error No.	Meaning
0	No I/O error has occurred.
10	(*RS-232-C only*). New output has been generated before previous output was finished being transmitted. In an RS-232-C configuration, the previous output will continue normally and the new output will be ignored (thus causing the error). (In an HP-IB configuration, this is not an error; any previous output that has not been read by the computer will be replaced by the new output.)
11	Invalid character received after first two characters (**ESC .**) in a device-control instruction.
12	Invalid character received while parsing a device-control instruction. The parameter containing the invalid character and all following parameters are defaulted.
13	Parameter out-of-range.
14	Too many parameters received. Additional parameters beyond the proper number are ignored; parsing of the instruction ends when a colon (normal termination) or the next **ESC** character (abnormal termination) is received. **NOTE:** The receipt of a character other than another parameter, a semicolon, or a colon will result in error 12 overwriting error 14. ■
15	(*RS-232-C only*). A framing error, parity error, or overrun error has been detected. The defective character is replaced by **DEL** (decimal code 127).
16	(*RS-232-C only*). The physical I/O buffer has overflowed. As a result, one or more characters have been lost; therefore, an HP-GL error will probably occur. The last valid character is replaced by **DEL** (decimal code 127).
17	(*RS-232-C only*). Transmit underrun. This can be caused by a baud rate mismatch between devices, or by excessive I/O activity in receive monitor mode.
18	I/O error of indeterminate cause.

Using the ESC . @ and ESC . E Instructions for Block I/O Error Checking (RS-232-C Only)

If you are using an RS-232-C configuration, you can also use the ESC . E instruction to check for transmission errors in a data block. First enable

ManualsPlus.com

block I/O error checking by setting bit 4 of the second parameter of the ESC.@ instruction to logic state "1" (decimal value 16). Then begin sending data blocks, following each with the ESC.E instruction. For each data block, if there is no I/O error (response to the ESC.E is 0), the data block is executed normally and a new data block begins. If the response to ESC.E indicates an I/O error, the plotter discards the entire data block. This allows you to correct the error and retransmit the data block, thus preventing errors in your plot. The following diagram illustrates block I/O error checking.

Block I/O Error Checking

Computer	Plotter	Comments
ESC.@;16: →		Enable block I/O checking.
Data block A →		Send a block of data. [Assume a character gets garbled (e.g., bad parity).]
ESC.E →		Any I/O errors?
	← 15 [TERM]	Parity, framing, or overrun error. At this point, the plotter discards the block because an error occurred.
Data block A →		Retransmit the block.
ESC.E →		Any I/O errors?
	← 0 [TERM]	No errors. Plotter executes block.
Data block B →		Send a block of data. [Assume a handshake character gets lost, and buffer overflows.]
ESC.E →		Any I/O errors?
	← 16 [TERM]	Buffer overflow. Plotter discards block because an error occurred.
Data block B →		Retransmit the block.
ESC.E →		Any I/O errors?
	← 0 [TERM]	No errors. Plotter executes block.

ManualsPlus.com

The Output Extended Status Instruction, ESC.O

USES: The ESC.O instruction outputs the plotter's extended status. Use this instruction to obtain immediate information about the current operating status of the plotter.

SYNTAX: **ESC.**O

RESPONSE: <DEC> [TERM]

EXPLANATION: The ESC.O instruction is similar to the HP-GL OS instruction in that they both output information about the plotter's status. However, the ESC.O instruction outputs more information, and it is executed immediately (instead of entering the buffer, as the OS instruction does).

The output response is the decimal equivalent value of a 16-bit immediate status word. The bits of the extended status word are defined in the following table.

Bit No.	Logic State	Decimal Value	Meaning
0	0	0	Manual paper-feed mode is selected on the front panel.
	1	1	Automatic paper-feed mode is selected on the front panel.
1	0	0	Current page is clean. (Clean means a pen has not been lowered).
	1	2	Current page is not clean. Set under the following conditions: • The plotter has been turned on (the plotter doesn't know whether the current page is clean, so it assumes the page is not clean). • A pen has been lowered. • An attempted paper load has failed (paper was not sensed).

(Table continued)

ManualsPlus.com

Bit No.	Logic State	Decimal Value	Meaning
2	0	0	Paper has not been loaded (from front panel, or a PG, AF, or AH instruction) since the last ESC . O instruction was executed. The ESC . O instruction resets this bit to 0.
	1	4	Paper has been loaded (from front panel, or a PG, AF, or AH instruction) since last ESC . O instruction was executed. This bit is set to 1 when the plotter is turned on.
3	0	0	I/O buffer is not empty.
	1	8	I/O buffer is empty and ready for data.
4 5	0 0	0	Remote state; processing HP-GL instructions.
4 5	1 0	16	View state; paper is loaded but graphics are suspended.
4 5	0 1	32	Not-ready state; paper is not loaded, and graphics are suspended.
6	0	0	Cover is lowered.
	1	64	Cover is raised.
7	0	0	**ENHANCED** mode is active.
	1	128	**STANDARD** mode is active.
8	0	0	Not used.
	1	256	Not used.

(Table continued)

ManualsPlus.com

Bit No.	Logic State	Decimal Value	Meaning
9	0	0	ESC.O, ESC.U, WD, or OK instruction has been executed. Execution of each of these instructions resets this bit to 0.
	1	512	Function key has been pressed while plotter is in keyboard mode.
10	0	0	Servo is functioning.
	1	1024	Servo is not functioning.
11–15	—	—	Not used.

The plotter's output response is the sum of the decimal equivalent values of all of the bits that are set to logic state 1. If you only care about certain bits being set, you can check those bits by using one of the methods described under Monitoring the Status Byte in Chapter 12. (For example, you might want to check bit 2 for a paper feed, or bit 9 for a function key being pressed in keyboard mode.) Or, you can easily determine the plotter's current status by the following process.

First, have your program read the output response. Then look at the decimal value column in the previous table and find the largest value that can be subtracted from the number returned in the output response. The bit corresponding to this value is set to the condition described in the table for logic state "1." Now subtract that value and look for the next largest value, and so on until the result is 0. This process is shown in the following example:

Plotter's output response:	518	
Subtract largest possible decimal equivalent value:	−512	(512 indicates a function key has been pressed; bit 9 is set to "1")
	6	
Subtract next value:	−4	(4 indicates a paper feed has occurred; bit 2 is set to "1")
	2	
Subtract next value:	−2	(2 indicates the current page has been plotted on; bit 1 is set to "1")
	0	(All other bits are set to "0"; refer to the previous table for the plotter's status when these bits are set to "0")

If your application requires that your program continually interrogate the plotter with ESC.O until a certain bit has been set, you should

ManualsPlus.com

cause the program to wait about 0.1 seconds between successive ESC.O instructions. (In BASIC, use the WAIT statement or an empty string, INPUT #1, N$.) The wait allows the plotter the time to execute other instructions, thus giving the plotter the opportunity to set the desired bit.

The Set Monitor Mode Instruction, ESC.Q

USES: The ESC.Q instruction enables or disables either parse monitor mode or receive monitor mode. Use this instruction to enable or disable a monitor mode from your program. You can also enable and disable both monitor modes from the front panel.

SYNTAX: **ESC**.Q (<DEC>):

Parameter	Range	Default
monitor mode	0–2	0

EXPLANATION: Executing ESC.Q without parameters (**ESC**.Q:) disables programmatic control of monitor mode. The parameters are interpreted as follows:

0 Disables programmatic control of monitor mode (default)

1 Enables parse monitor mode

2 Enables receive monitor mode

The monitor modes are a debugging aid for program development. In order to use the monitor modes, you must have both a computer and a terminal, and the appropriate cables to connect them to the plotter. You will use the computer to transmit HP-GL and device-control instructions to the plotter (via the plotter's **HP-IB** or **COMPUTER/MODEM** port). The plotter will execute the instructions as usual, but it will also copy data to the display of the terminal (via the plotter's **TERMINAL** port). In this way, depending on the form of monitor mode that you use, you can see exactly what the plotter is receiving, parsing, and outputting.

The difference between the two modes is *when* data is transmitted to the terminal, as described next:

- In the parse monitor mode, all characters are transmitted to the terminal as they are parsed by the plotter. Thus, the terminal displays whatever the plotter is currently doing (for example, if the plotter is selecting pen 2, the display shows SP2;).
- In the receive monitor mode, all characters are transmitted to the terminal as they are received by the plotter. Thus, the terminal displays whatever the plotter will be doing later when the received data is parsed.

ManualsPlus.com

To enable one of the monitor modes, you can use the front-panel function keys or you can use the ESC.Q instruction in your program. (You can also enable or disable monitor mode with bits 2 and 3 of the ESC.@ instruction, although this method is not recommended because it is more difficult than using the ESC.Q instruction.)

NOTE: If you enable a monitor mode from the front panel, you cannot disable it or change it from your program. If you enable a monitor mode from your program, you can toggle between the modes or disable monitor mode either from the front panel, or from your program. Any time you change the state of the monitor mode from the front panel, the new state overrides any state currently set in your program. For this reason, the front-panel display always reflects what has been set by the front panel, and *not* the current programmatic state. ■

The following two subsections present information relevant to HP-IB and RS-232-C configurations, respectively. Following these paragraphs are illustrations of data flow in the two monitor modes.

Notes for HP-IB Configurations

To use a monitor mode, you must have a computer *and* a terminal. Connect the computer to the plotter's **HP-IB** port with any standard HP-IB cable, and connect the terminal to the plotter's **TERMINAL** port with the terminal's RS-232-C cable. Then set the plotter's front-panel **BAUD** function key to match the baud setting on the terminal.

All data coming to the plotter from the terminal (except for the break signal) will be ignored. The plotter responds to a terminal-generated break signal by clearing the I/O buffer, thus halting current operations. If the plotter is in addressable mode, any plotter output responses are sent to both the computer and the terminal. If the plotter is in listen-only mode, output responses are not possible. Refer to Chapter 15 for information on addressable and listen-only modes.

Notes for RS-232-C Configurations

Before reading these paragraphs, you should know the meanings of such terms as programmed-on, standby, bypass, stand-alone, and eavesdrop. These modes and configurations are all discussed in Chapter 16.

The plotter must be in the programmed-on state before you can enable a monitor mode. (That is, both standby mode and bypass mode must be off.) If the plotter is in local mode, enabling a monitor mode places the plotter in remote mode.

While the plotter is in a monitor mode, all data coming to the plotter from the terminal will be ignored (except for the break signal) in order to assure that the terminal will not take part in any handshake process between the

ManualsPlus.com

plotter and the computer. The effect of the terminal-generated break signal depends on whether the plotter is operating in the eavesdrop configuration, or the stand-alone configuration. In eavesdrop, a break signal causes the plotter to clear the I/O buffer and return to the programmed-off state. In stand-alone, a break signal has no effect on the plotter; the plotter ignores it.

If you enable the *receive* monitor mode, you should be aware that the the setting of the front-panel **DUPLEX** function key affects the flow of output responses, as follows. (Refer also to the data flow illustrations shown after these paragraphs.)

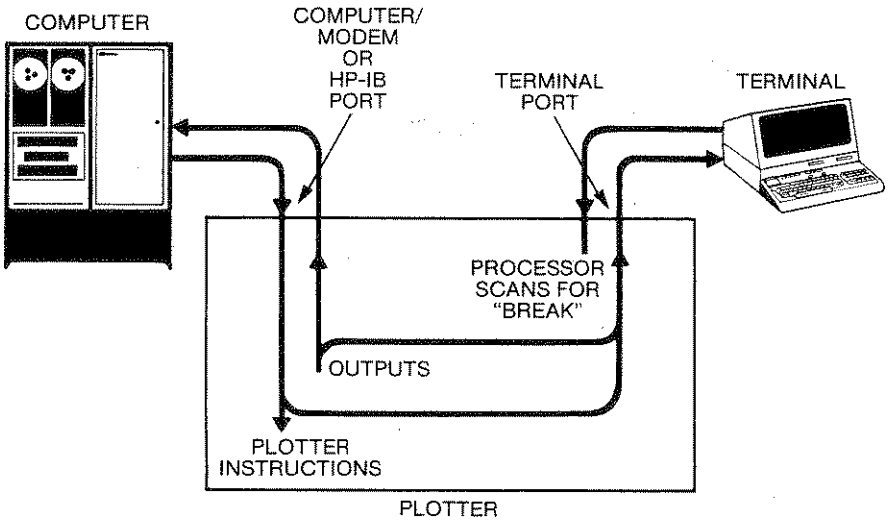
- If the plotter is set to **FULL DUPLEX** and the computer is working in an echo-plex environment, plotter output responses are sent to the computer, but *not* to the terminal. In full duplex operations, plotter responses to the computer are echoed to the terminal; if the plotter explicitly sent output responses to the terminal, the terminal would receive the responses twice (the explicit response and the echoed response).
- If the plotter is set to **HALF DUPLEX**, plotter output responses are sent to both the computer and the terminal. In half duplex operations, plotter responses are not echoed, so the plotter must explicitly send the responses to the terminal.

(In *parse* monitor mode, the setting of the **DUPLEX** switch is not important. This is because only *parsed* data from the plotter's buffer is transmitted to the terminal. If the computer echoes a plotter output response, this response never gets echoed to the terminal because it is not parsed by the plotter. Therefore, the plotter must explicitly send any output response to the terminal.)

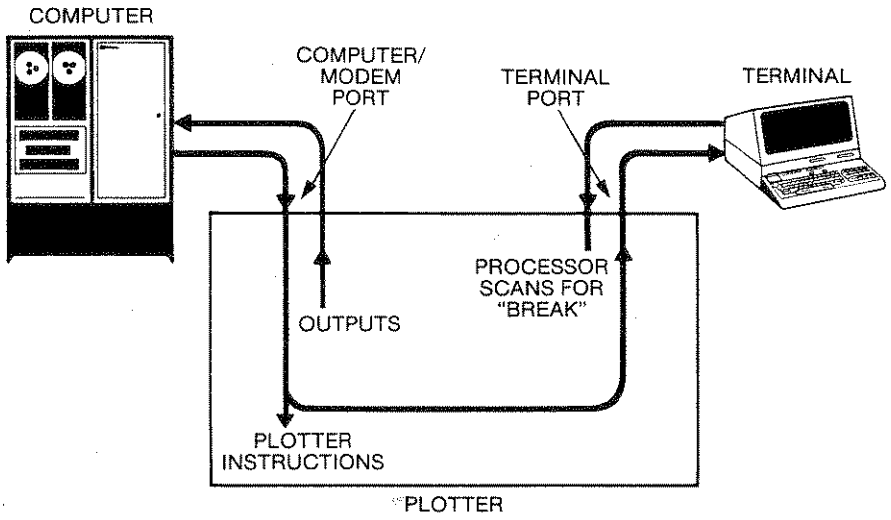
Data Flow in the Monitor Modes
(Any Interface Configuration)

The following diagrams illustrate data flow for the two monitor modes.

ManualsPlus.com



Parse Mode (HP-IB or RS-232-C) and Receive Mode (Half-Duplex RS-232-C)



Receive Mode (Full-Duplex RS-232-C)

The Abort Graphics Instruction, ESC.K

USES: The ESC.K instruction aborts any partially parsed HP-GL instruction and clears the remaining instructions from the buffer. Use this instruction as part of an initialization sequence when starting a new program or to terminate plotting of the HP-GL instructions remaining in the buffer.

ManualsPlus.com

SYNTAX: ESC.K

EXPLANATION: The ESC.K instruction aborts any partially parsed HP-GL instruction, but permits the instruction being executed to finish, with this exception: the currently executing vector is allowed to be completed, but all other scheduled vectors are aborted. This condition is particularly evident if ESC.K is executed while the plotter is in the process of completing label, arc, circle, polygon, or line type instructions that contain multiple vector moves, or if a VS instruction has specified a slow velocity.

In addition, all pending HP-GL instructions in the buffer are discarded, and the parser is reset. In an RS-232-C configuration, any data entered since block I/O error checking was enabled (with ESC.@) is also aborted.

The Reset Instruction, ESC.R

USES: The ESC.R instruction resets certain I/O conditions to the states that exist when the plotter is first turned on. Use this instruction in an initialization sequence when starting a new program.

SYNTAX: ESC.R

EXPLANATION: The ESC.R instruction aborts any currently executing device-control instruction, aborts any partially parsed HP-GL instruction, resets the parser, defaults to any handshake protocol established on the front panel, clears all buffers, and defaults the allocations of the five buffers in the configurable graphics memory. Executing ESC.R is equivalent to executing this four-instruction sequence:

ESC.J ESC.K ESC.P:ESC.T:

For more detailed information on what each of these instructions does, refer to their individual descriptions. The abort device control instruction, ESC.J, and the set handshake mode instruction, ESC.P, are described in Chapter 16. The abort graphics instruction, ESC.K, and the allocate configurable graphics memory instruction, ESC.T, are described earlier in this chapter.

NOTE: After executing the ESC.R instruction, you should execute an output buffer size when empty instruction, ESC.L. This ensures that all conditions have been reset before any subsequent instruction can be parsed. In this application of the ESC.L instruction, the numerical value of the buffer size is not important. However, your program should read the output response before sending more data, in order to avoid potential errors. You might also wish to execute ESC.L before executing ESC.R. Refer to the hints under the ESC.T instruction for more information. The ESC.T and ESC.L instructions are described earlier in this chapter. ■

ManualsPlus.com

The Output Identification Instruction, ESC.A

USES: The ESC.A instruction outputs the plotter's model number. Use this instruction to obtain the plotter's model number when you want to know which plotter is currently on-line.

SYNTAX: **ESC**.A

RESPONSE: <ASC>,<DEC> [TERM]

EXPLANATION: The ESC.A instruction is similar to the HP-GL OI instruction in that they both output the plotter's model number. However, the ESC.A instruction is executed immediately (instead of entering the buffer, as the OI instruction does), and ESC.A also outputs the firmware revision level.

The model number is always 7550A, output in a character string. The firmware revision level is output as integers. The model number and firmware revision level are separated by a comma.

The End Flush Mode Instruction, ESC.U

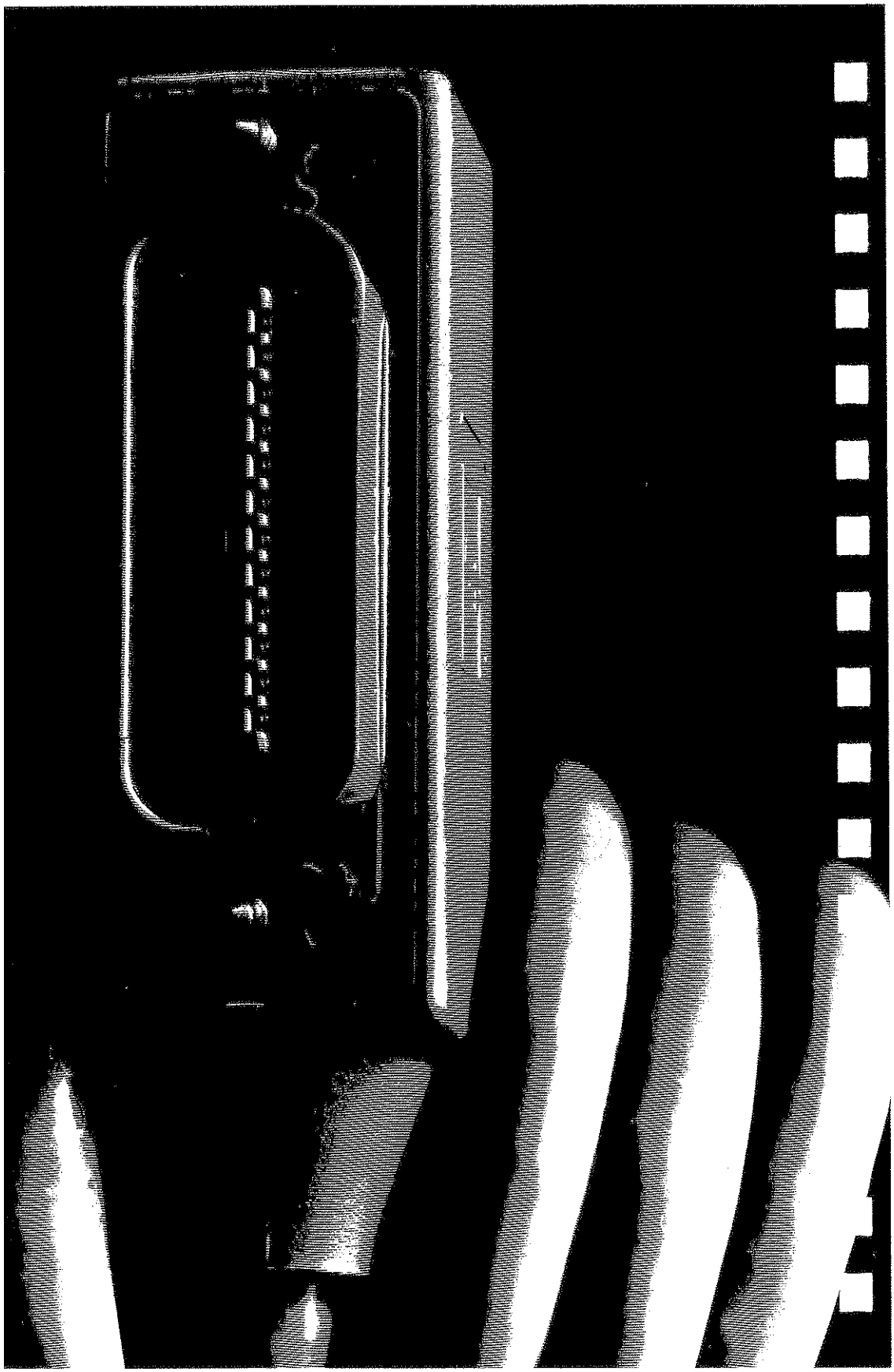
USES: The ESC.U instruction ends flush mode. Use this instruction in spooling applications to end flush mode, thus allowing the plotter to begin parsing HP-GL instructions again.

SYNTAX: **ESC**.U

EXPLANATION: Flush mode is enabled when the front-panel **ESCAPE** function key is pressed, but is not triggered until the OG instruction has been executed (refer to the OG instruction in Chapter 10). Once triggered, flush mode causes all incoming HP-GL data to be discarded. This is a useful feature in spooling applications, where it is important for the plotter and application program to be able to respond to an operator-initiated "escape."

The ESC.U instruction causes the plotter to stop discarding HP-GL data, and resets bit 9 of the extended status word (refer to the ESC.O instruction, described earlier in this chapter). After execution of ESC.U, HP-GL instructions can again be parsed.

ManualsPlus.com



ManualsPlus.com

Chapter 15
HP-IB Interfacing

What You'll Learn in This Chapter

This chapter is for HP 7550 owners who choose to use the Hewlett-Packard Interface Bus (HP-IB). In this chapter you will learn how to use the HP-IB addressing technique to ensure that the plotter receives only the data (such as HP-GL) intended for it. You will learn about addressing the plotter as a talker, a listener, or in listen-only mode. Program examples will show you how to send and receive data using a variety of HP computers. In most cases, this will be all the information you need to use HP-IB interfacing.

For those who need to understand how the interface works, this chapter explains the HP-IB lines and operations. For detailed information, read the section on HP-IB functions and their implementation on the plotter. Also included is a description of serial and parallel polling.

What Is the HP-IB?

The Hewlett-Packard Interface Bus (HP-IB) provides for mechanical, electrical, timing, and data compatibility between all devices adhering to the ANSI/IEEE 488-1978 standard. If you connect several HP-IB devices and assign them unique "addresses," you can be sure they will be able to communicate with each other.

For most applications, you will probably only need to understand the addressing concept, described next in this chapter. However, if you wish to know more about how the HP-IB works, you can read further in this chapter, beginning with the section titled An HP-IB Overview. The HP-IB conforms to the ANSI/IEEE 488-1978 standard. For complete details, you should refer to this standard. Another document you might find helpful is the Tutorial Description of the HP-IB (Part No. 5952-0156). The information in this chapter covers the specific manner in which the standard is implemented by the HP 7550 plotter.

ManualsPlus.com

Addressing the Plotter

The HP-IB uses an addressing technique to ensure that each device on the bus receives only the data intended for it. Using this addressing technique, alternate devices can be instructed to talk (send) or listen (receive). More than one device can listen at the same time, but only one device can be designated as a talker at any given time.

There are basically two modes of addressing the plotter: addressable and listen-only modes. In addressable mode, the plotter can function as a talker or as a listener, depending on the commands it receives from the computer. In listen-only mode, the plotter hears all activity on the bus, but it cannot talk.

Addressable Mode

The addressing technique on most HP desktop computers requires assigning a "select code" to the HP-IB interface and an "address code" to each peripheral device. The select code and address code are combined in certain program statements to address different devices. In the HP BASIC "PRINTER IS 705" statement, 7 is the interface select code and 05 is the address code of the plotter. The plotter will then receive the information sent by "PRINT" statements. The plotter can be set to 31 different HP-IB addresses, ranging from 0 through 30, or listen-only mode (described in the following section). Select addresses using the front-panel **HP-IB** function key. Each HP-IB interface can have as many as 15 devices connected to it, set to different specific address codes. The plotter is factory set to an address of 05. Be sure your plotter's address is different from your computer or other peripherals.

The HP Touchscreen (150) computers do not specify the plotter address or a select code in programs. They use the Microsoft® BASIC OPEN statement — OPEN "O",1,"PLT". Thereafter, send information using "PRINT #1" statements (without quotes). The "1" indicates the file you have opened for output, not the address of the plotter.

NOTE: When using the plotter with an HP desktop computer, do not use address 21; it is reserved for the desktop computer's address. ■

Listen-Only Mode

Activate listen-only mode by using the **HP-IB** function key. The **LISTEN ONLY** setting follows the address **30** setting. In listen-only mode, the plotter does not have an address, but listens to all data transmitted on the bus. The plotter cannot be placed in a talker-active state and will not respond to a serial or parallel poll. Listen-only mode is useful in a system that has no controller but instead has a dedicated talker, such as a magnetic tape drive or other mass storage unit, transmitting information to the plotter. Also, when more than one plotter is connected to the

ManualsPlus.com

HP-IB, you can plot on all plotters simultaneously by setting all but one to listen-only mode. Use the one plotter that isn't in listen-only mode to do the talking for all of the plotters.

Notes on Addressing Protocol

In order to communicate effectively with the plotter, it is important that you understand the HP-IB addressing protocol of your computer. Therefore, you may wish to review this aspect of your computer.

Some computers can use high-level languages (such as BASIC, FORTRAN, Pascal, and COBOL) with high-level input/output (I/O) statements. In this case, the addressing procedure (unlisten, talk, listen) is taken care of by the computer's internal operating system and need not be of concern to you. With these high-level I/O statements, you may not be able to control some of the other bus functions. If your system fits this description, you do not need to read the following paragraph. Skip to the next section, Sending and Receiving Data.

Some computers must use low-level I/O statements to address devices on the HP-IB bus. If your computer uses such statements, you'll need to direct the talking, listening, and unlistening activities. For example, to tell a plotter with an address of 05 to listen, you would send the ASCII character % along with the proper control line. To tell the same plotter to talk, you would send the ASCII character E. Refer to the HP-IB Overview section for more details and a table of address characters and their ASCII and octal values.

Sending and Receiving Data

This section provides program examples to illustrate ways of sending and receiving data between the plotter and various HP computers. These programs also show you a variety of ways to address the plotter. Follow the instructions for your particular computer and run the programs. If your computer isn't listed, or you prefer to use another programming language, use the programs provided as an example when writing your own programs.

Computer-to-Plotter Examples

Transmitting data from a computer to the plotter is typically accomplished using statements such as WRITE, WRITELN, PRINT, PRINT#, and OUTPUT. Running the following programs will cause the plotter to label the identity of the computer sending data, beginning at the X,Y coordinates 1000, 2000. The programs send both character strings and numeric data as variables, constants, and literals. (For more information on sending variables, refer to Chapter 4 for numeric data and to Chapter 7 for character strings.)

ManualsPlus.com

HP 9000, Series 200, Models 216, 226, and 236
(Previously HP 9816, 9826, and 9836)

In the following BASIC program, the "PRINTER IS 705" statement specifies that the data in the PRINT statements is to be sent to the device on interface 7 at address 5. The interface select code is factory-set to 7 on the HP Series 200 computers, and the address is set to 5 on the HP 7550. Using the front-panel function keys, make sure your plotter is set to address **05**. The CHR$(3) function, used in line 60, specifies the label terminator **ETX**.

```
10   PRINTER IS 705
20   A$=" SENDING DATA"
30   B=216
40   Y=2000
50   PRINT "SP1;PA1000,";Y;";"
60   PRINT USING "K";"LBHP ";B;A$;CHR$(3);"SP0;"
70   END
```

Plotted result: HP 216 SENDING DATA

When using HP Model 226 or 236 computers, substitute the appropriate model number in line 30. For example, when using an HP Model 226, line 30 should read: B = 226.

HP Series 100, Model 150

When using the HP 150, you must use "PLT" as the filename for the OPEN statement, as shown in line 10. Using the front-panel function keys, make sure your plotter is set to address **05**. When you enter line 10 of this BASIC program, make sure you type the letter "O" (for output), rather than a zero. The CHR$(3) function, used in line 60, specifies the label terminator **ETX**.

```
10   OPEN "O",1,"PLT"
20   A$=" SENDING DATA"
30   B=150
40   Y=2000
50   PRINT #1, "SP1;PA1000,";Y;";"
60   PRINT #1, "LBHP ";B;A$+CHR$(3)+"SP0;"
70   END
```

Plotted result: HP 150 SENDING DATA

HP 9000, Series 500, Models 520, 530, and 540

BASIC Example

In the following BASIC program, the "PRINTER IS 205" statement specifies that the data in the PRINT statements is to be sent to the device on interface 2 at address 5. Make sure the interface card is in slot #2 of your computer. Using the front-panel function keys, make sure your plotter is set to address **05**. The CHR$(3) function, used in line 60, specifies the label terminator **ETX**.

ManualsPlus.com

```
10    PRINTER IS 205
20    A$=" SENDING DATA"
30    B=520
40    Y=2000
50    PRINT "SP1;PA1000,";Y;";"
60    PRINT USING "K";"LBHP ";B;A$;CHR$(3);"SP0;"
70    END
```

Plotted result: HP 520 SENDING DATA

When using HP Model 530 or 540 computers, substitute the appropriate model number in line 30. For example, when using an HP 530, line 30 should read: B = 530.

FORTRAN Example

The following program is written in FORTRAN 77 and assumes that the system has been configured so that a special file "/dev/plotter" is created by the superuser (system administrator) using the mknod command. Refer to the HP-UX System Administrator Manual (Part No. 97089-90047), for details regarding the mknod command. Line numbers as given are only necessary for the format statements.

The integer field width specification, I4, is adequate to plot to the coordinate Y = 2000. You may wish to use a field width of 5 or 6 to allow for pen positions up to the hard-clip limits of the plotter. A field larger than six characters would only be necessary for scaled data greater than 999 999 or less than −999 999. The variable IETX is set to 3, which is the decimal code for the label terminator **ETX**.

```
                CHARACTER*14 TEXT
                INTEGER ILU,IETX,B,Y
                TEXT=" SENDING DATA"
                ILU=20
                IETX=3
                B=520
                Y=2000
                OPEN(UNIT=ILU,FILE='/dev/plotter')
                WRITE(ILU,1000) "PU;SP1;PA1000,",Y,";"
1000            FORMAT(X,A14,I4,A1)
                WRITE(ILU,1001) "LBHP ",B,TEXT,IETX
1001            FORMAT(X,A5,I4,A14,R1)
                STOP
                END
```

Plotted result: HP 520 SENDING DATA

When using HP Model 530 or 540 computers, substitute the appropriate model number. For example, when using an HP Model 530, the seventh line should read: B = 530.

ManualsPlus.com

Plotter-to-Computer Examples

Transmitting data from the plotter to the computer is typically accomplished using statements such as READ, READLN, INPUT, and ENTER. The following examples of obtaining output responses from the plotter using various HP computers are intended to illustrate the necessity for understanding the I/O statement protocol implemented on your computer. Running these programs will cause the plotter to move the pen to the X,Y coordinates 1000,1000 and then output the current pen position, pen status (0 = pen up), and plotter identifier string to the computer. (Refer to Chapter 13 for more information on obtaining plotter output responses.)

HP 9000, Series 200, Models 216, 226, and 236

(Previously HP 9816, HP 9826, and HP 9836)

In the following BASIC program, the "PRINTER IS 705" statement specifies that the data in the PRINT statements is to be sent to the device on interface 7 at address 5. Similarly, the "ENTER 705" statement specifies that data is to be received from the device on interface 7 at address 5. The interface select code is factory-set to 7 on the HP Series 200 computers, and the address is set to 5 on the HP 7550. Using the front-panel function keys, make sure your plotter is set to address **05**.

```
10    OUTPUT 705; "PA1000,1000;OC;"
20    ENTER 705;A,B,C
30    OUTPUT 705;"OI;"
40    ENTER 705;A$
50    DISP A,B,C,A$
60    END
```

Displayed current pen position and identification:

```
1000        1000        0           7550A
```

HP Series 100, Model 150

Because of an interfacing problem in the HP-IB, the HP 150 cannot be used to obtain output information from the plotter.

ManualsPlus.com

BASIC Example

In the following BASIC program, the "PRINTER IS 205" statement specifies that the data in the PRINT statements is to be sent to the device on interface 2 at address 5. Similarly, the "ENTER 205" statement specifies that data is to be received from the device on interface 2 at address 5. Make sure the interface card is in slot #2 of your computer. Using the front-panel function keys, make sure your plotter is set to address **05**.

```
10    PRINTER IS 205
20    PRINT "PA1000,1000;OC;"
30    ENTER 205;A,B,C
40    PRINT "OI;"
50    ENTER 205;A$
60    DISP A,B,C,A$
70    END
```

Displayed current pen position and identification:

1000 1000 0 7550A

FORTRAN Example

The following program is written in FORTRAN 77 and assumes that the system has been configured so that the special file "/dev/plotter" is created by the superuser (system administrator) using the mknod command. Refer to the HP-UX System Administrator Manual (Part No. 97089-90047), for details regarding the mknod command. Line numbers as given are only necessary for the format statements.

The READ statement in the seventh line is formatted to read two 4-digit integers (1000,1000) and one 1-digit integer (0), separated by commas. The READ statement in the eleventh line is formatted to read a character string of five characters (7550A).

ManualsPlus.com

```
      CHARACTER*8 ID
      INTEGER A,B,C
      ILU=20
      OPEN(UNIT=ILU,FILE='/dev/plotter')
      WRITE(ILU,1000) "PA1000,1000;OC;"
1000  FORMAT(X,A15)
      READ(ILU,2000) A,B,C
2000  FORMAT(I4,x,I4,x,I1)
      WRITE(ILU,3000) "OI;"
3000  FORMAT(X,A3)
      READ(ILU,4000) ID
4000  FORMAT(A5)
      WRITE(6,5000) A,B,C,ID
5000  FORMAT(X,3I5,2X,A5)
      STOP
      END
```

Displayed current pen position and identification:

1000 1000 0 7550A

An HP-IB Overview

The purpose of the Hewlett-Packard Interface Bus (HP-IB) is to provide for mechanical, electrical, timing, and data compatibility between all devices adhering to the ANSI/IEEE 488-1978 standard. The interface functions for each system component are performed within the component, so only passive cabling is needed to connect the system. The cables connect all peripherals, controllers, and other components of the system in parallel to the signal lines.

There are 10 interface functions which provide the physical capability to communicate via HP-IB. Not all devices are capable of implementing all functions; check the documentation for your computer and other system components. The functions implemented by the HP 7550 are listed later in this section, following the general descriptions of HP-IB lines and operations.

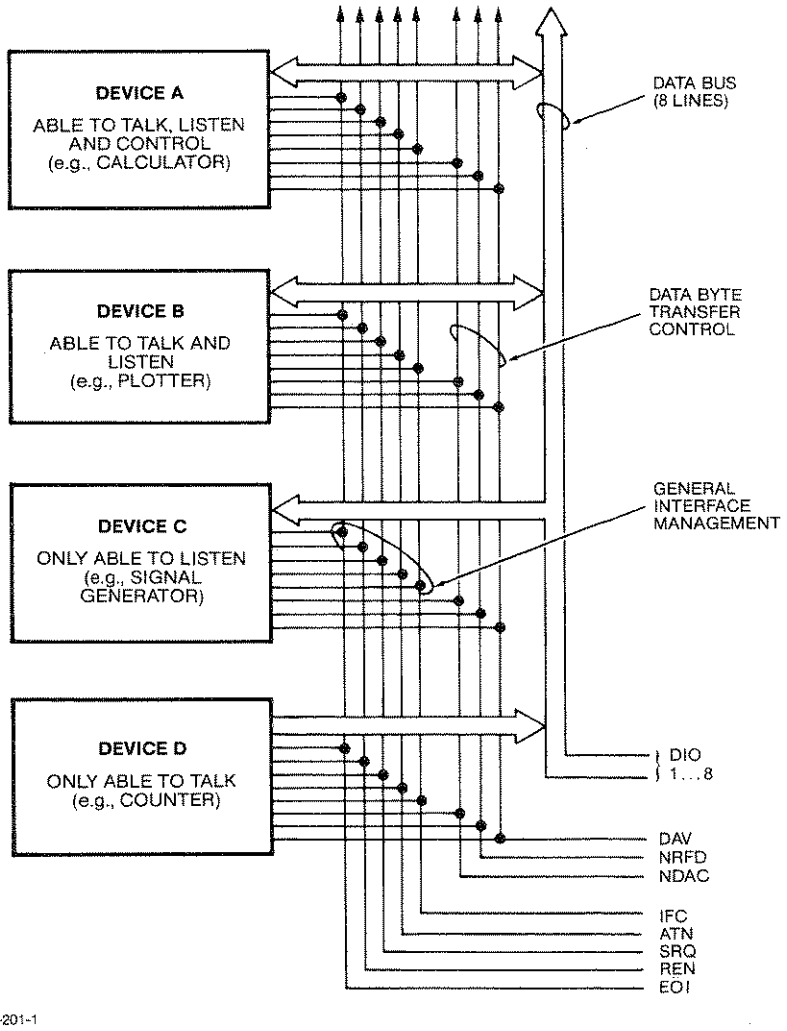
Refer to the ANSI/IEEE 488-1978 standard for a description of all of the HP-IB interface functions. This standard defines each interface function in terms of state diagrams that express all possible interactions. Another document you might find helpful is the Tutorial Description of the Hewlett-Packard Interface Bus (Part No. 5952-0156).

HP-IB Lines

The HP Interface Bus transfers data between the components of a system on 16 signal lines. The eight data I/O lines (DIO1 through DIO8) are reserved for the transfer of data in a byte-serial, bit-parallel manner. Data

ManualsPlus.com

transfer is asynchronous, coordinated by the three handshake lines: data valid (DAV), not ready for data (NRFD), and not data accepted (NDAC). The other five lines are for management of bus activity. See the following figure.



HP-IB Signal Lines

Devices connected to the bus may be talkers, listeners, or controllers. The controller dictates the role of each of the other devices by setting the ATN (attention) line true and sending talk or listen addresses on the data lines. Addresses are set into each device at the time of system configuration either by switches built into the device, by jumpers on a PC board, or, as for the HP 7550, by front-panel controls. While the ATN line is true, all devices must listen to the data lines. When the

ManualsPlus.com

ATN line is false, only devices that have been addressed will actively send or receive data. All others ignore the data lines.

Several listeners can be active simultaneously, but only one talker can be active at a time. Whenever a talk address is put on the data lines (while ATN is true), all other talkers will automatically be unaddressed.

Information is transmitted on the data lines under sequential control of the three handshake lines (DAV, NRFD, and NDAC). No step in the sequence can be initiated until the previous step is completed. Information transfer can proceed as fast as devices can respond, but no faster than allowed by the slowest device presently addressed as active. This permits several devices to receive the same message byte concurrently.

The ATN line is one of the five bus management lines. When ATN is true, addresses and universal commands are transmitted using the ASCII 7-bit code, on only seven of the data lines. When ATN is false, any code of 8 bits or less, understood by both talker and listener(s), may be used.

Setting the IFC (interface clear) line true places the interface system in a known quiescent state.

The REN (remote enable) line is used by the system controller with the Remote, Local, and Clear Lockout/Set Local capabilities to select either local or remote control of each device. The HP 7550 does not implement this feature.

Any active device can set the SRQ (service request) line true. This indicates to the controller that a device on the interface bus wants attention, such as a plotter that has detected an error and needs attention.

The EOI (end or identify) line is used by a device to indicate the end of a multiple-byte transfer sequence. When a controller sets both the ATN and EOI lines true, each device capable of a parallel poll indicates its current status on the DIO (data input/output) line assigned to it.

HP-IB Operations

The operation of the interface is generally controlled by one device equipped to act as a controller. The interface transmits a group of functions, which act as commands, to direct the other devices on the bus in carrying out their functions of talking and listening.

The controller has two ways of sending interface information. Information can be transmitted in a multiline fashion, over the eight data lines and three handshake lines; or in a uniline configuration, over the five individual lines of the management bus. The bus functions serve several purposes:

- Addresses or talk and listen commands select the devices that will transmit and accept data. They are all multiline messages.

ManualsPlus.com

- Universal commands cause every device equipped to do so to perform a specific interface operation. They include multiline messages and three uniline commands: interface clear (IFC), remote enable (REN), and attention (ATN).
- Addressed commands (also referred to as primary commands) are similar to universal commands, except that they affect only those devices that are addressed and are all multiline commands. A device responds to an addressed command, however, only after it has been addressed as a talker or listener.
- Secondary commands are multiline messages that are always used in series with an address, universal command, or addressed command to form a longer version of each. Thus, they extend the code space when necessary.

To address a device, the controller uses seven of the eight data lines. This allows devices using the ASCII 7-bit code to act as controllers. Five bits are available for addresses, and a total of 31 allowable addresses are available in one byte. If all secondary commands are used to extend this into a two-byte addressing capability, 961 addresses become available (31 allowable addresses in the second byte for each of the 31 allowable in the first byte).

Addressing Sequences

One of the first things you must consider when directly controlling the HP-IB is addressing. Following is a typical addressing sequence.

<Unlisten Command> <Talk Address> <Listen Addresses>

This sequence is made up of three major parts which serve the following purposes:

1. The unlisten command is the universal bus command with a character code of "?". It unaddresses all listeners. After the unlisten command is transmitted, no active listeners remain on the bus.
2. The talk address designates the device that is to talk. A new talk address automatically unaddresses the previous talker.
3. The listen addresses designate one or more devices that are to listen. A listen address adds the designated device as listener along with other addressed listeners.

This basic addressing sequence simply states who is to talk to whom. The commands are implemented by putting data on the bus and setting the ATN line true. The unlisten command ("?") plays a vital role in this sequence. It is important that a device receive only the data that is intended for it.

ManualsPlus.com

When a new talk address is transmitted in the addressing sequence, the previous talker is unaddressed. Therefore, only the new talker can send data on the bus and there is no need to routinely use an untalk command in the same manner as the unlisten command.

To tell a computer at address 21 to talk and a plotter at address 05 to listen, the controller (usually the computer) sets the ATN line true and sends the following sequence over the data lines:

?U%

where ? — tells all devices on the bus to unlisten
U — designates the device at address 21 as the talker
% — designates the device at address 05 as the listener.

Following is a table of ASCII characters and their decimal and octal equivalent values.

Address Codes

ASCII Characters		Equivalent Values	
Listen	**Talk**	**Decimal (Front-Panel Setting)**	**Octal**
SP	@	0	0
!	A	1	1
”	B	2	2
#	C	3	3
$	D	4	4
%	E	5	5
&	F	6	6
’	G	7	7
(	H	8	10
)	I	9	11
*	J	10	12
+	K	11	13
,	L	12	14
–	M	13	15

PRESET →

ManualsPlus.com

Address Codes (Continued)

ASCII Characters		Equivalent Values	
Listen	**Talk**	**Decimal (Front-Panel Setting)**	**Octal**
.	N	14	16
/	O	15	17
0	P	16	20
1	Q	17	21
2	R	18	22
3	S	19	23
4	T	20	24
5	U	21	25
6	V	22	26
7	W	23	27
8	X	24	30
9	Y	25	31
:	Z	26	32
;	[	27	33
<	\	28	34
=	]	29	35
>	^	30	36
?	_	LISTEN ONLY	37

RESERVED FOR HP DESKTOP COMPUTER ADDRESS

RESERVED FOR UNIVERSAL UNLISTEN COMMAND

Interface Functions

Interface functions provide the physical capability to communicate via the HP-IB. There are 10 interface functions, plus two special cases of the controller function. All HP-IB devices do not implement all functions. The table on the next page lists the functions implemented on the HP 7550. A dash (—) indicates that the HP 7550 does not implement that function.

ManualsPlus.com

HP-IB Interface Functions

Mnemonic	Interface Function Name	Plotter Implementation
SH	Source Handshake	SH1
AH	Acceptor Handshake	AH1
T	Talker (or TE = Extended Talker)*	T6
L	Listener (or LE = Extended Listener)*	L3
SR	Service Request	SR1
RL	Remote Local	—
PP	Parallel Poll	PP1, PP2 or PP0**
DC	Device Clear	DC1
DT	Device Trigger	—
C	Any Controller	—
C_N	A Specific Controller (for example: C_A, C_B...)	—
C_S	The System Controller	—

*Extended Talkers and Listeners use a two-byte address. Otherwise, they are the same as Talker and Listener.

**PP1 (if address > 8), PP2 (if address < 8), PP0 (if listen-only mode).

Bus Capabilities

To enable bus capabilities, devices must be capable of implementing specific functions. Each device in a system may be designed to use only the messages that are applicable to its purpose in the system. It is important to be aware of the HP-IB functions implemented on each device in your HP-IB system to ensure the operational compatability of the system. For example, you may want the plotter to set the service request line true to indicate a need for service. In this case, the controlling device must be capable of implementing the Controller function in order to receive the service request message from the plotter. Refer to the documentation for the devices you'll be using to determine which functions they implement.

The following table describes the full implementation of all HP-IB system capabilities and the functions required to implement them.

ManualsPlus.com

HP-IB Bus Capabilities and Functions

Bus Capability	Functions Required Sender Function → Receiver Function(s) (Support Functions)
Data enables data, such as HP-GL, to be sent between a controller and one or more devices.	T → L* (SH, AH)
Trigger enables listening device(s) to perform a device-dependent action when addressed.	C → DT* (L, SH, AH)
Clear sets either the listening device(s) or all of the devices on the bus to their predefined device-dependent states.	C → DC* (L, SH, AH)
Remote causes listening device(s) to switch from local front-panel control to remote program control.	C_S → RL* (SH, AH)
Local clears Remote from listening device(s) and returns device(s) to local front-panel control.	C → RL* (L, SH, AH)
Local Lockout prevents a device operator from manually inhibiting program control.	C → RL* (SH, AH)
Clear Lockout/Local removes all devices on the bus from Local Lockout and reverts them to Local.	C_S → RL*
Service Request is sent by a device to notify the controller that it needs some type of interaction. The service request is cleared when the Status Byte is sent by the device requesting service.	SR* → C

*Since more than one device can receive (or send) this message simultaneously, each device must have the function which is indicated by an *.

ManualsPlus.com

HP-IB Bus Capabilities and Functions (Continued)

Bus Capability	Functions Required Sender Function → Receiver Function(s) (Support Functions)
Status Byte represents the status of a device on the bus. Bit 6 indicates whether the device sent a Service Request; remaining bits indicate operational conditions defined by the device. Sent from a talking device in response to a serial poll from a controller.	T → L* (SH, AH)
Status Bit represents the operational conditions of several devices on the bus. Each device responds on a particular line which represents a bit of the byte, indicating a device-dependent condition. Typically sent in response to a parallel poll.	PP* → C
The Status Bit can also be used by a controller to specify the particular bit and logic level that a device will respond with when a parallel poll is performed. More than one device can respond on the same bit.	
Pass Control transfers the bus management responsibilities from the active controller to another controller.	C_A → C_B (T, SH, AH)
Abort is sent by the system controller to unconditionally assume control of the bus from the active controller. Terminates all bus communications, without implementing a Clear.	C_S → T, L, *C

*Since more than one device can receive (or send) this message simultaneously, each device must have the function which is indicated by an *.

ManualsPlus.com

Serial and Parallel Polling

Serial and parallel polling are the processes used by the computer to determine the state of devices on the HP-IB bus. The conditions which will cause a service request to be sent to the computer by the plotter in a serial poll, and the plotter to respond positively to a parallel poll, are defined by the input mask instruction, IM, explained in Chapter 13.

The Serial Poll

A serial poll enables the computer to learn the status or condition of devices on the interface bus. The serial poll is so named because the computer polls devices one at time, rather than all at once.

The plotter will respond to a serial poll by sending the status byte as described under The Output Status Instruction, OS, in Chapter 13. The S-mask parameter of the input mask instruction, IM, is used to specify which status byte conditions will send a service request. Unless you change the S-mask value from the default setting of 0, the plotter will not send a service request.

When any of the conditions designated by the S-mask are true, bit position 6 of the status byte will be set to 1. Bit position 6 will be set to 0 after a serial poll has occurred or the condition causing the initial request no longer exists. After a serial poll has been made, the service request won't be sent again until the bit in the status byte which originally caused the service request to be sent has been cleared and the condition which caused the original service request occurs again.

A computer performs a serial poll by issuing a serial poll enable and addressing the plotter to talk. It must then terminate the serial poll by issuing a serial poll disable. Refer to your computer's documentation to determine whether or not your system has serial poll capability and for the necessary commands. During a serial poll, a device must be instructed to talk and the computer to listen. Therefore, a serial poll cannot be executed when the plotter is in listen-only mode.

The Parallel Poll

If your system implements parallel polling, the parallel poll may be the fastest method of determining which device on the bus needs service. In one operation, the parallel poll determines which device (with parallel poll capability) is requesting service. If the response to the poll is affirmative, then a serial poll can be conducted to obtain the device's specific status. Not all computers have parallel poll capability. Refer to your computer's documentation to determine whether or not your system has parallel poll capability and for the necessary commands.

ManualsPlus.com

If the HP 7550's address is 0 through 7, the address determines which data line is used, as shown in the following table. With address settings greater than 7, a response will not be sent unless the plotter has been remotely configured by the computer. In this case, the response is sent on the data line specified by the computer. The plotter will respond positively to a parallel poll only if the conditions specified in the P-mask are satisfied and parallel poll response is enabled. The P-mask parameter of the input mask instruction, IM, is used to specify which status byte conditions will result in a positive response to a parallel poll. The plotter's response to a parallel poll is to set the appropriate data line to a logical 1.

Plotter Address	Parallel Poll Bit		HP-IB Data Line Number
	Position	Value	
0	7	128	8
1	6	64	7
2	5	32	6
3	4	16	5
4	3	8	4
PRESET → 5	2	4	3
6	1	2	2
7	0	1	1

To execute a parallel poll, the controller sets the ATN and EOI lines true. The computer reads the eight data lines and determines from these lines which device on the bus is responding positively. The computer then sends the parallel poll disable command.

It is important to remember that the plotter will not set the appropriate data line to a logical 1 unless the P-mask value has been changed from the default value of 0 and some condition included in the new P-mask value is true. The plotter does not respond to a parallel poll in listen-only mode.

ManualsPlus.com

Notes

ManualsPlus.com

ManualsPlus.com

Chapter 16
RS-232-C/CCITT V.24 Interfacing

What You'll Learn in This Chapter

This chapter pertains to the RS-232-C interface. It describes the different ways the plotter, computer, and a terminal may be physically configured in an RS-232-C or CCITT V.24 system. The various modes of operation for the plotter, a description of the mechanical interface, communication requirements, and a step-by-step checklist are provided to help you set up your system.

In addition, a tutorial description of the three predefined handshake methods, and details on how to select them from the front panel or under program control is provided. Information on controlling the format of plotter output, and a detailed description of the device-control instructions you may use to tailor your own handshake method are also included. These device-control instructions follow the syntax conventions presented in Chapter 14. Note that device-control syntax differs from HP-GL syntax.

NOTE: All information in this chapter applies equally to RS-232-C, RS-422-A, and CCITT V.24 interfaces, except where noted. For purposes of simplicity, all are referred to as RS-232-C. ■

Device-Control Instructions Covered

ESC.P Set Handshake Mode Instruction

ESC.M Set Output Mode Instruction

ESC.N Set Extended Output and Handshake Mode Instruction

ESC.H Set Handshake Mode 1 Instruction

ESC.I Set Handshake Mode 2 Instruction

ESC.J Abort Device Control Instruction

ESC.Y or ESC.(Plotter-On Instruction

ESC.Z or ESC.) Plotter-Off Instruction

ManualsPlus.com

Terms You Should Understand

Programmed-On/**BYPASS OFF** — In the programmed-on state, the plotter will accept and interpret all data as plotter instructions. (Refer to Modes of Operation later in this chapter). If the plotter is set up in a stand-alone configuration, the default power-on state is programmed-on, and can *only* be changed with the front-panel **BYPASS** function key. In an eavesdrop configuration, the programmed-on state can be entered manually with the front-panel **BYPASS OFF** function key, or from a program with the plotter-on instruction, ESC.Y or ESC.(. While programmed-on, in either the eavesdrop or local stand-alone configurations, the plotter will monitor the terminal port for a break signal. The plotter will interpret this break signal as an instruction to flush the I/O buffer and, in an eavesdrop configuration, will return the plotter to the programmed-off state.

Programmed-Off/**BYPASS ON** — When the plotter is programmed-off, it allows all data to travel through it between the computer and terminal until another plotter-on instruction is received. This state is most useful in an eavesdrop configuration, where the plotter shares one port to the computer with a terminal. In this configuration, the programmed-off state can be established by selecting the **BYPASS ON** function key, or sending a plotter-off instruction, ESC.Z or ESC.). In a stand-alone configuration, the programmed-off state can *only* be established with the **BYPASS ON** function key.

Introduction

Basically, it is very easy to connect the plotter in a computer system and select one of the predefined handshake methods (Xon-Xoff, enquire/acknowledge, or hardwire). A step-by-step checklist is provided later in this chapter; it outlines how to correctly interface the plotter in a given system, and how to set up communications and select a handshake method.

First, you should decide how you want to physically configure your computer, plotter, and terminal. The first sections of this chapter describe the possible configurations. Then, as described later, you can choose one of the standard predefined handshake methods, using the front-panel function keys, or under program control with the ESC.P instruction.

NOTE: Any settings you make at the front panel are sensed immediately when entered and will override any temporary conditions established by a software program. Also, the following front-panel interfacing functions are stored in the plotter's continuous memory: eavesdrop/stand-alone, handshake, modem, direct, full/half duplex, parity, 7-bit/8-bit, and baud rate. Thus, the plotter will retain whatever settings you make at the front panel, even when you turn the plotter off and on again. ■

ManualsPlus.com

The plotter goes into serial (RS-232-C) mode if the first data it receives after power is applied comes through one of the RS-232-C ports. All data that subsequently enters through the HP-IB port will be ignored until the plotter is turned off and on again.

The information on serial interfacing in this manual applies specifically to the HP 7550 plotter. For more detailed information on interfacing and handshaking, refer to HP Plotter Note No. 6 (Part No. 5953-4163).

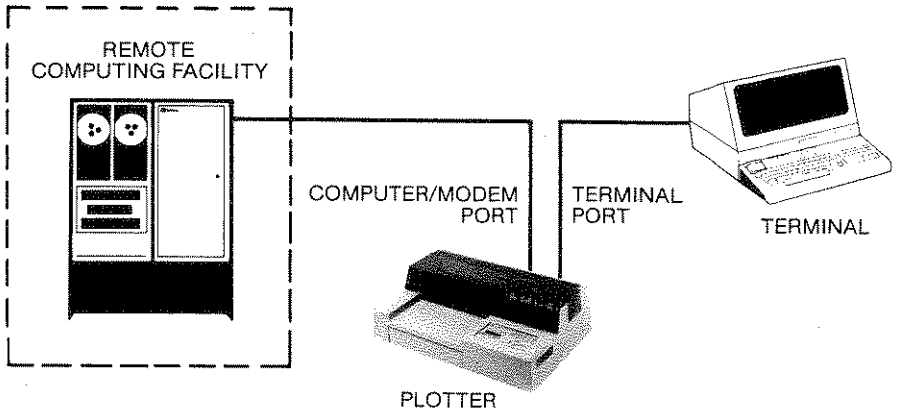
Plotter Configurations

The two plotter configurations, eavesdrop and stand-alone, are described next. Choose the configuration that applies to the physical setup of your computer, terminal (if any), and plotter.

Eavesdrop Configuration

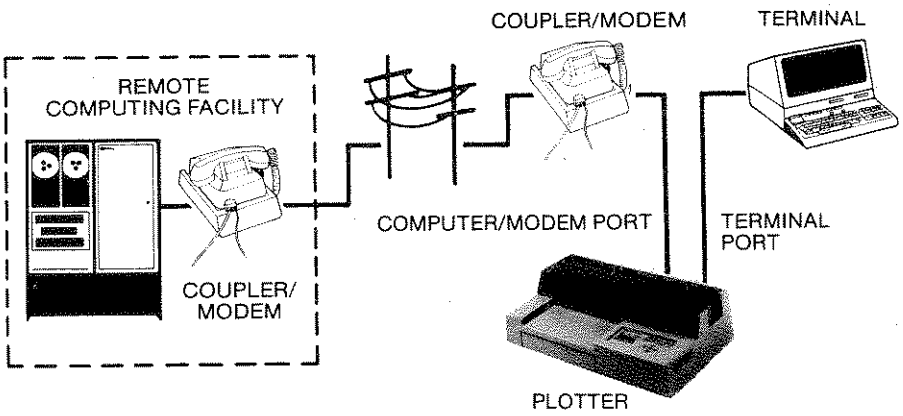
In an eavesdrop configuration, the plotter is connected in series between a computer and another peripheral (usually a terminal) as shown in the next illustrations. The term "eavesdrop" comes from the fact that when the plotter is turned on, it listens in or "eavesdrops" on the RS-232-C interface line and allows data to travel through it between the computer and the terminal. An advantage to this configuration is that the terminal and plotter can share one line to the computer.

The plotter will continue to eavesdrop on the line until a plotter-on instruction, ESC . Y or ESC . (, is received. After the plotter "hears" this instruction, it will go into the programmed-on state and interpret all data as either HP-GL or device-control instructions. If the plotter receives a plotter-off instruction, ESC . Z or ESC .), it will return to the programmed-off state, and again, pass on all data. (The programmed-on and off states can also be established by the **BYPASS OFF** and **BYPASS ON** function keys, respectively.)



Plotter, Computer, and a Terminal in an Eavesdrop Configuration

ManualsPlus.com



Plotter, Computer, and a Terminal
in an Eavesdrop Configuration with Modems

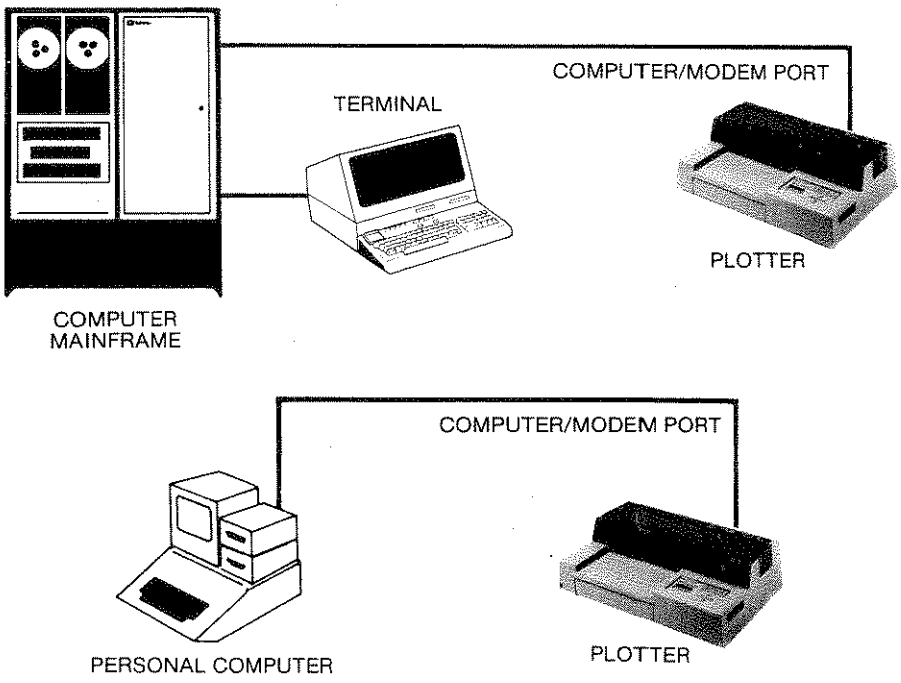
To set up the plotter in an eavesdrop configuration, set the **STANDALONE/EAVESDROP** front-panel function key to **EAVESDROP**, then connect the computer and terminal to the connectors on the rear panel of the plotter. There are two types of connectors: one male connector for the computer/modem cable and one female connector for the terminal cable.

Stand-Alone Configuration

In the stand-alone configuration, the plotter is connected directly to a computer and is usually adjacent to it. Diagrams of this type of configuration for both large computer systems and personal computers appear on the next page. In a stand-alone configuration, the default operating state is programmed-on. The programmed-off state can only be set from the front panel (with the **BYPASS ON** function key) since the plotter is connected directly to the computer and never needs to ignore data that might be passing through it (as it does in an eavesdrop configuration).

To set up the plotter in a stand-alone configuration, set the **STANDALONE/EAVESDROP** front-panel function key to **STANDALONE**, then connect the computer to one of the connectors on the rear panel of the plotter. You can use either the male or female connector, depending on the cable required for your computer.

ManualsPlus.com



Stand-Alone Configurations with a Mainframe or Personal Computer

Modes of Operation

In either the eavesdrop or stand-alone configuration, there can be three modes of operation: either remote, local, or standby. Each of these modes can be selected with the front-panel function keys.

Remote Mode

In remote mode, the plotter will accept data for plotting only from the male port on the rear panel of the plotter. The male port is labeled **COMPUTER/MODEM**. If the computer or modem echoes data, **FULL** duplex should be set and an echo terminate character defined with the ESC.M instruction. Otherwise, select **HALF** duplex operation at the front panel.

NOTE: In remote mode, simultaneous transmission of plotter output and terminal-generated data will result in interleaved and garbled data transmission to the computer (or computer to terminal if in local mode). ■

Local Mode

In local mode, the plotter will accept data for plotting only from the female port on the rear panel of the plotter. The female port is labeled **TERMINAL**. If the terminal connected to the **TERMINAL** port expects any data that it sends to be echoed back to it, the **DUPLEX** function key should

ManualsPlus.com

be set to **FULL**. The plotter will then supply this echo response. If the terminal is set for local echo, set the plotter to **HALF** duplex so that the plotter will not echo data sent by the terminal.

NOTE: One of the standard handshaking methods, hardwire handshaking (DTR, pin 20), is not possible in local mode. ■

Standby Mode

In standby mode, the plotter will ignore all data sent to it no matter what configuration (eavesdrop or stand-alone) or state (programmed-on or programmed-off) it is in. In standby mode, the plotter remains in the programmed-off state, either passing data through in the eavesdrop configuration, or ignoring all data in the stand-alone configuration. The standby mode is useful as a debugging aid in eavesdrop configurations that are having difficulties communicating through the plotter, or when transmitting binary data between the computer/modem and terminal.

A Description of Plotter Modes in the Eavesdrop and Stand-Alone Configurations

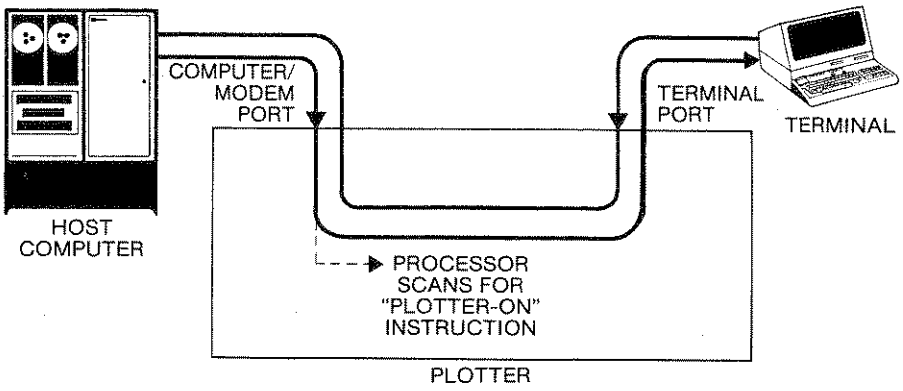
Your system configuration and the mode of operation you select from the front panel affect the operating state of the plotter. Following are functional descriptions for various combinations of modes of operation and the operating states for the eavesdrop and stand-alone configurations.

Remote Eavesdrop Mode, Programmed-Off

When the front-panel function keys are set for remote eavesdrop operation, the plotter can be programmed-on or programmed-off either from the front panel or from a program.

In the programmed-off state, the default power-on condition, the plotter passes data between the computer and the terminal as shown in the following diagram. The plotter will respond only after a plotter-on instruction, ESC.(or ESC.Y, from the computer, or when the front-panel **BYPASS** function key is set to **OFF**.

ManualsPlus.com



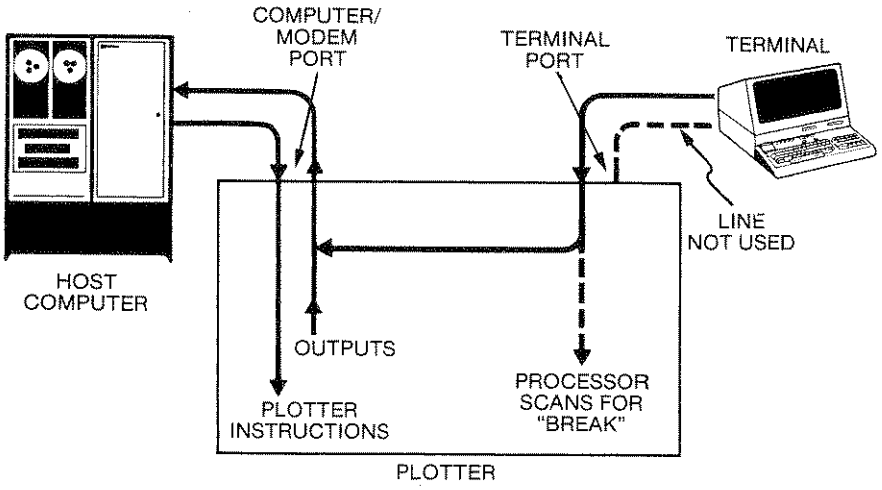
Plotter in Remote Eavesdrop Mode, Programmed-Off

Remote Eavesdrop Mode, Programmed-On

In the programmed-on state, the plotter operates in response to instructions received from the computer as shown in the following diagram. When the plotter instructions request output, it is provided as shown. The communication channel from the terminal to the computer, through the plotter, is maintained to provide operator access to the computer.

The plotter's processor monitors the channel from the terminal to the computer for a terminal-generated break signal. The plotter will interpret this break signal as an instruction to flush the I/O buffer and return the plotter to the programmed-off state. A break signal of equal duration is retransmitted to the computer.

Note that simultaneous transmission of plotter outputs and terminal-generated data will result in interleaved data transmission to the computer.

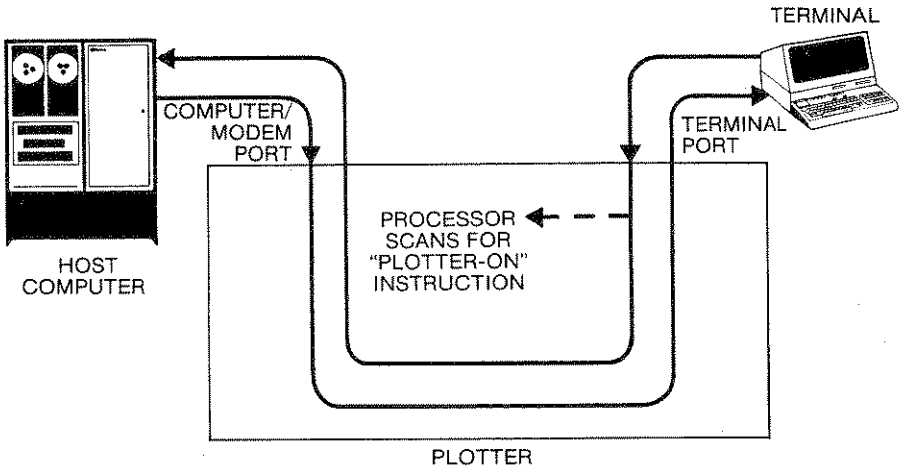


Plotter in Remote Eavesdrop Mode, Programmed-On

ManualsPlus.com

Local Eavesdrop Mode, Programmed-Off

Local eavesdrop operation is very similar to remote eavesdrop mode, except that data is received from the **TERMINAL** port, not the **COMPUTER/MODEM** port. In the local programmed-off mode, the plotter will respond only to a plotter-on instruction, ESC . (or ESC . Y, from the terminal or when the front-panel **BYPASS** function key is set to **OFF**. Terminal-generated data is displayed on the terminal only if the data is echoed from a host computer or the terminal is set for local echo. This mode is normally used where the host computer must service many terminal modes and it is desirable to download to a smart terminal to reduce computer-connect time. Data routing in this mode is shown in the following diagram.

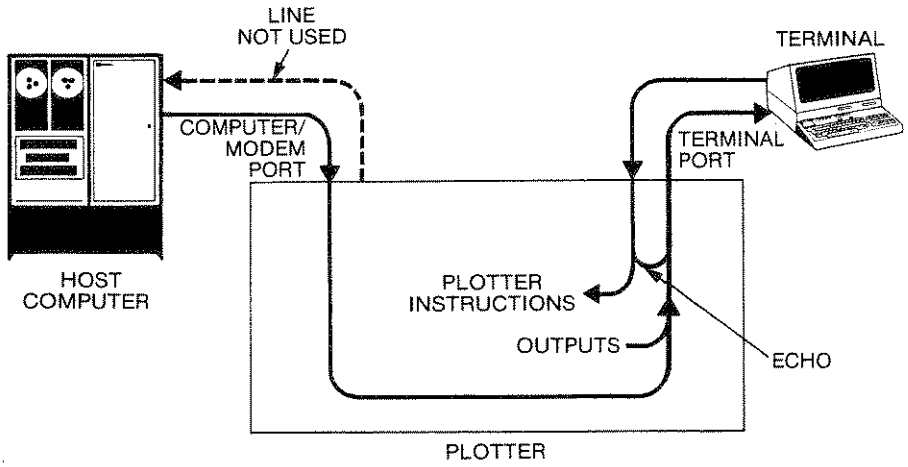


Plotter in Local Eavesdrop Mode, Programmed-Off

Local Eavesdrop Mode, Programmed-On

In the local programmed-on mode, the plotter operates in response to instructions received from the terminal as shown in the following diagram. When the plotter instructions request output, the output is sent only to the terminal, as shown. In **FULL** duplex, the plotter instructions are echoed back to the terminal, and in **HALF** duplex, the echo is suppressed.

ManualsPlus.com



Plotter in Local Eavesdrop Mode, Programmed-On (Full Duplex)

Remote Stand-Alone Mode

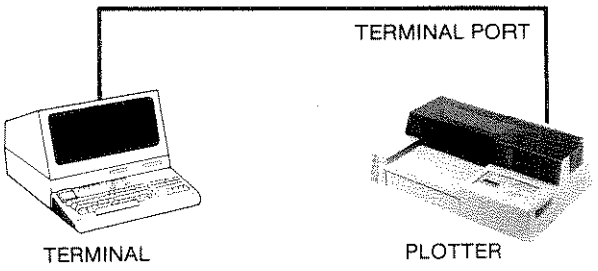
When the front-panel function keys are set for remote stand-alone operation, the data source is the **COMPUTER/MODEM** port. The plotter may not be programmed-on or programmed-off from a program.

In the programmed-on state, the power-on default condition, the plotter interprets all data from the computer as plotter instructions.

Local Stand-Alone Mode

In local stand-alone operation, the data source is the **TERMINAL** port. The plotter can only be programmed-on or off with the front-panel **BYPASS** function key. If **FULL** duplex is selected, all terminal-generated characters are echoed back to the terminal.

In the programmed-on state, the plotter monitors the channel from the terminal to the computer for a terminal-generated break signal. The plotter will interpret this break signal as an instruction to flush the I/O buffer.



Plotter in Local Stand-Alone Mode

ManualsPlus.com

Automatic Disconnection Modes

Two modes are available for automatic disconnection at the end of an RS-232-C/CCITT V.24 session conducted over phone lines. Disconnection is achieved when no one is present to manually hang up the phone. The two modes are switched/datex-line disconnection and leased-line disconnection. The only way to leave either mode is to turn the plotter off and on again.

Switched/Datex-Line Disconnection Mode

In this mode, the CTS and DSR (CB and CC) lines control the DTR line and plotter bypass mode. As long as the CTS and DSR lines are high, the DTR line is high. When either the CTS or DSR line goes low, the DTR line goes low; this causes a disconnection, and the plotter switches to **BYPASS ON**. To activate this mode, press the pen select **5** key while turning on the plotter.

Leased-Line Disconnection Mode

In this mode, the CTS, DSR, and DCD (CB, CC, and CF) lines control the DTR line and plotter bypass mode. As long as these three lines are high, the DTR line is high. If any of the three lines goes low, the DTR line goes low; this causes a disconnection, and the plotter switches to **BYPASS ON**. To activate this mode, press the pen select **6** key while turning on the plotter.

Direct/Modem Mode

The front-panel function keys may be used to select the direct or modem mode. The direct mode is most commonly used, although with personal computers a special cable may be required. The modem mode can be enabled and used to hold off data transmission from the plotter if either of the CTS or DSR (CB or CC) lines goes low.

Direct or modem mode may also be set under program control with the ESC . @ instruction, described in Chapter 14.

Check your computer documentation to determine whether the DSR and CTS lines are monitored on your system, or whether direct mode and a specially-wired cable should be used.

ManualsPlus.com

Mechanical Interface and Connector Pin Allocations

RS-232-C/CCITT V.24 Interface Implementation

The plotter interfaces to the RS-232-C communications lines through standard 25-pin connectors, one female labeled **TERMINAL**, and one male labeled **COMPUTER/MODEM**. The plotter is compatible with RS-232-C and CCITT V.24 protocols.

When a hardwire handshake method is used, the Data Terminal Ready (DTR) line (pin 20 on the **COMPUTER/MODEM** connector) is used to signal whether space is available in the logical I/O buffer for more data.

RS-422-A Interface Implementation

The EIA RS-422-A standard defines the electrical characteristics of balanced voltage digital interface circuits for serial data communication. The differential voltage (balanced) nature of this interface allows a much greater interconnecting cable distance compared to the RS-232-C standard, which is for an unbalanced interface.

Each signal in RS-422-A is carried on a pair of differentially driven lines. Thus, electrical noise that appears equally on both lines will be ignored since only the difference is being used.

In the HP 7550, the Transmitted Data and Received Data signals on both the **COMPUTER/MODEM** port and the **TERMINAL** port are duplicated in RS-422-A. These lines are provided on the two 25-pin connectors. On the **TERMINAL** port, they occupy pin positions 9, 10, 18, and 25. On the **COMPUTER/MODEM** port, they occupy pin positions 9, 10, 18, and 3.*

A 5-wire RS-422-A interface can be built consisting of a Send Data pair, a Receive Data pair, and Signal Common. Hewlett-Packard has standardized such an interface, and the HP 7550 can be configured to operate in this environment by using the 25-to-5-pin adapter cable (Part No. HP 17855A).

An enquire/acknowledge, Xon-Xoff, or other type of software handshake must be used with the adapter cable, because no RS-422-A hardwire handshake lines are provided.

Connector Pin Allocations

Details of the connector pin allocations, including pin numbers, signal directions, and signal levels for the RS-232-C, CCITT V.24, and RS-422-A interface specifications, are shown in the following tables.

*This is true for HP 7550's with a serial number larger than 09801.

RS-232-C/CCITT V.24 Interface on **COMPUTER/MODEM** Port

Pin No.	Function	RS-232-C	CCITT V.24	Signal Direction and Level
1	Protective Ground	AA	(none)	Not applicable
2	Transmitted Data (TD)	BA	103	Data from plotter High = SPACE = "0" = +12 V Low = MARK = "1" = −12 V
3	Received Data (RD)	BB	104	Data to plotter High = SPACE = "0" = +3 V to +25 V Low = MARK = "1" = −3 V to −25 V
4	Request to Send (RTS)	CA	105	Signal from plotter High = ON = +12 V Low = OFF = −12 V
5	Clear to Send (CTS)	CB	106	Signal to plotter High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
6	Data Set Ready (DSR)	CC	107	Signal to plotter High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
7	Signal Ground (SGND)	AB	102	Not applicable

ManualsPlus.com

8	Data Carrier Detect (DCD)	CF	109	Signal to plotter High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
*17	External Baud Rate Input	DD	115	Signal to plotter High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
20	Data Terminal Ready (DTR)	CD	108.2	Signal from plotter High = ON = +12 V Low = OFF = −12 V
23	Data Signal Rate Selector	CH/CI	—	Signal from plotter Always High = ON = +12 V

*An external clock input to pin 17 of the connector allows operation of the plotter at any intermediate baud rate up to 9600 baud. Both the receiver (RRC) and transmitter (TRC) clocks will operate at the same clock rate. Requirements for the clock signal are as follows:

1. The clock frequency must be 16 times the desired baud rate.
2. The baud rate must not exceed 9600.
3. The duty cycle of the clock pulse should be close to 50%.
4. The clock pulse must be a logic "on" of $+2\ V < V < 25\ V$ and a logic "off" of $-25\ V < V < +0.8\ V$ (3.5 kΩ input impedance).
5. Care should be taken to keep the transmission lines as short as possible to minimize transmission line reflection noise.

ManualsPlus.com

RS-232-C/CCITT V.24 Interface on **TERMINAL** Port

Pin No.	Function	RS-232-C	CCITT V.24	Signal Direction and Level
1	Protective Ground	AA	(none)	Not applicable
2	Transmitted Data (TD)	BA	103	Data to plotter High = SPACE = "0" = +3 V to +25 V Low = MARK = "1" = −3 V to −25 V
3	Received Data (RD)	BB	104	Data from plotter High = SPACE = "0" = +12 V Low = MARK = "1" = −12 V
4	Request to Send (RTS)	CA	105	Signal to plotter High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
5	Clear to Send (CTS)	CB	106	Signal from plotter High = ON = +12 V Low = OFF = −12 V
6	Data Set Ready (DSR)	CC	107	Signal from plotter High = ON = +12 V Low = OFF = −12 V

ManualsPlus.com

ManualsPlus.com

7	Signal Ground (SGND)	AB	102	Not applicable
8	Data Carrier Detect (DCD)	CF	109	Signal from plotter High = ON = +12 V Low = OFF = −12 V
17	External Baud Rate Input	DD	115	Signal to plotter High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V
20	Data Terminal Ready (DTR)	CD	108.2	Signal to plotter High = ON = +3 V to +25 V Low = OFF = −3 V to −25 V

RS-422-A Interface on **COMPUTER/MODEM** Port

Pin No.	Function	RS-422-A	Signal Direction and Level	Adapter Cable Pin No.
9	Send Data −	SD.A	Signal from plotter High = SPACE = "0" = +5 V Low = MARK = "1" = 0 V	2
10	Send Data +	SD.B	Signal from plotter High = MARK = "1" = +5 V Low = SPACE = "0" = 0 V	4
18	Receive Data +	RD.B	Signal to plotter High = MARK = "1"** Low = SPACE = "0"**	5
*3	Receive Data −	RD.A	Signal to plotter High = SPACE = "0"** Low = MARK = "1"**	3
7	Signal Common	SG	Not applicable	1

*This is true for HP 7550's with a serial number larger than 09801.

**The absolute voltage level in each input does not determine a logic state by itself. Instead, the logic state is determined by the voltage level difference between the two inputs. A 0.2-volt differential is necessary to ensure that the logic state is recognized properly. Refer to the EIA standard on RS-422-A for additional details.

ManualsPlus.com

RS-422-A Interface on **TERMINAL** Port

Pin No.	Function	RS-422-A	Signal Direction and Level	Adapter Cable Pin No.
*2 & 9	Send Data −	SD.A	Signal to plotter High = SPACE = "0"** Low = MARK = "1"**	2
10	Send Data +	SD.B	Signal to plotter High = MARK = "1"** Low = SPACE = "0"**	4
18	Receive Data +	RD.B	Signal from plotter High = MARK = "1" = +5 V Low = SPACE = "0" = 0 V	5
25	Receive Data −	RD.A	Signal from plotter High = SPACE = "0" = +5 V Low = MARK = "1" = 0 V	3
7	Signal Common	SG	Not applicable	1

*Pins 2 and 9 are tied together, since pin 2 of the RS-232-C interface shares the same logic circuitry as pin 9 of the RS-422-A interface.

**The absolute voltage level in each input does not determine a logic state by itself. Instead, the logic state is determined by the voltage level difference between the two inputs. A 0.2-volt differential is necessary to ensure that the logic state is recognized properly. Refer to the EIA standard on RS-422-A for additional details.

ManualsPlus.com

ManualsPlus.com

Transmission Errors

Transmission errors occur when communication between the computer and plotter is incomplete or does not conform to what is expected or required by either device.

Transmission errors include:

- Framing Error — The plotter does not detect a valid stop bit at the end of every character.
- Parity Error — The plotter does not detect the expected parity (odd or even). This error is inhibited if the plotter **PARITY** function key is set to **OFF**.
- Overrun Error — A character writes over another character before it is placed in the buffer or processed.
- Buffer Overflow Error — The plotter receives more bytes of data than it has space for in the I/O buffer.

When the plotter detects a framing, parity, or overrun error, it sets error 15. This error generally indicates that the communication problem is hardware-related (e.g., wrong parity selection, incompatible or incorrectly set baud rates, etc.) If a parity error occurs, the bad character is replaced with the ASCII delete character **DEL** (decimal code 127).

When the plotter detects an I/O buffer overflow, it sets error 16 and the last character received is replaced with the **DEL** character. The last HP-GL data that caused the overflow will be lost. Error 16 generally indicates an improperly established handshake protocol.

When an error is set, a message appears on the front-panel display. You can also determine the error number using the ESC.E instruction in your program. A complete list of RS-232-C errors is included with the discussion of the ESC.E instruction in Chapter 14; this list also appears in Appendix B.

NOTE: A buffer overflow condition could also cause an HP-GL error 7 to occur. In this case, you could determine the error number using the OE instruction (discussed in Chapter 13). ■

Setting Up the Plotter: a Checklist

The following paragraphs outline the choices that must be made to correctly interface the plotter in a given RS-232-C system.

Determine System Configuration

Determine whether you will be operating in an eavesdrop or stand-alone configuration as shown in the section on Plotter Configurations earlier

ManualsPlus.com

in this chapter. Select the appropriate configuration with the front-panel function keys. The appropriate operating state of default programmed-off or programmed-on will automatically be set.

Select Data Source

If your plotter will be receiving data from a program on a host computer, select remote mode. Remote mode is selected by default when the power is turned on. If you will be sending instructions directly from the **TERMINAL** port, select the local mode. If you select the standby mode, the plotter cannot be programmed-on and ignores all data.

Check Hardware Connections

Check the Operation and Interconnection Manual to make sure you have the required cables, and connect the plotter as shown for your chosen configuration. Use the appropriate cable for your system when connecting to the **COMPUTER/MODEM** port. Use the cable supplied with your terminal when connecting to the **TERMINAL** port.

CAUTION

When data is received from either the **TERMINAL** or **COMPUTER/MODEM** port, nothing should be attached to the **HP-IB** port.

Set Baud Rate and Stop Bits

Determine the baud rate at which your computer sends data and use the **BAUD** rate function key to set the plotter to the same rate. When shipped from the factory, the plotter is configured to automatically verify or generate two stop bits at 110 and lower baud rates, and one stop bit for all baud rates greater than 110.

Set Parity and Data Bits

Determine if parity checking is used on your system and use the front-panel function key to set parity **ODD**, **EVEN**, or **OFF**, accordingly. The parity bit must be set to match in the computer, plotter, and terminal.

Set 7 or 8 data bits to correspond to your system. If you are not sure of how many data bits your system requires, try the following procedure:

- If your system uses no parity, start with **8** data bits. If error 15 occurs when you transmit data to the plotter, reset to **7** data bits.
- If your system uses odd or even parity, start with **7** data bits. If error 15 occurs when you transmit data to the plotter, check for proper parity, or change to **8** data bits.

ManualsPlus.com

Set Direct or Modem Mode

Select the **DIRECT** function key on the front panel if the Data Set Ready (DSR) and Clear to Send (CTS) lines are not used on your system. Most systems use this mode. Note that a special cable may be required to interface the plotter with certain systems. See the section on interface pin allocations, described previously in this chapter.

Select the **MODEM** function key on the front panel if your system requires that the DSR and CTS lines must be set high before the plotter transmits data to the host. This mode may also be enabled from a program with the ESC.@ instruction, described in Chapter 14.

Set Echo On or Off

Set the duplex mode to **FULL** on the front panel to echo all received data back to the terminal in local mode, or if data is echoed back from the computer in remote mode. There is no echo when **HALF** duplex is selected.

If the plotter is set to remote mode, **FULL** duplex is selected, and an echo terminate character is defined (typically a line-feed character) after the plotter's output, then all input is ignored until the echo terminate character is received. The ESC.M instruction can be used to change the echo terminate character if the plotter seems to be ignoring data sent to it after an output request.

Select a Handshake Method

Determine which handshake method your system uses. You may select one of three predefined handshake methods with the front-panel function keys, or under program control with the ESC.P instruction. The three predefined handshake methods are Xon-Xoff, enquire/acknowledge, and hardwire. They are explained in the section called Understanding HP 7550 Predefined Handshake Methods next in this chapter.

If your system requires a special format for handshaking, different from the conditions established with the predefined methods, you can tailor your own handshake method. Tailoring a handshake method is described later in this chapter.

If you encounter errors, or communication between the computer and plotter is not what you expected, refer to the section called Transmission Errors earlier in this chapter.

Understanding HP 7550 Predefined Handshake Methods

The purpose of handshaking is to assure correct and complete data transfer. The computer and the plotter must transfer information to one

ManualsPlus.com

another in such a way that data will not be lost. The plotter uses a buffer, called the logical input/output buffer, to adjust for the rates at which data is received and processed.

The plotter is capable of using any one of three predefined handshake methods to prevent buffer overflow and the resulting loss of data. The computer system's capabilities and requirements dictate which handshake method is appropriate. The predefined handshake methods are:

- Xon-Xoff Handshake — This method is initiated by the plotter. It can be used if the computer system follows an Xon-Xoff protocol, where control characters are transmitted from the plotter to the computer to indicate when the logical I/O buffer is full, and when to send more data.
- Enquire/Acknowledge Handshake — This handshake method is initiated by the computer system. It is used in Hewlett-Packard systems and is so named because the ASCII characters **ENQ** and **ACK** may be used as handshake characters.
- Hardwire Handshake — This method uses a physical wire, pin 20 of the RS-232-C connector, to control handshaking. It can be used if the computer system can or does monitor pin 20 (the DTR line). However, the hardwire handshake cannot be used when the plotter is in local mode or when the special RS-422-A interface is being used.

Choosing a Handshake Method

Any of the standard handshake methods can be selected using front-panel function keys or under program control with the ESC.P instruction. In most cases, you can use either of these methods to easily choose a predefined handshake method and not worry about the intricacies of handshake protocols.

If you need to change the standard values used for the turnaround delay, output trigger character, echo terminate character, or output initiator in any of these standard handshake methods, simply use the ESC.M instruction. For more details, refer to the section on Controlling Plotter Output later in this chapter.

In certain applications, however, you may need to tailor a special handshake method to meet the requirements of your system. In this case, refer to the section on Tailoring Your Own Handshake later in this chapter. The handshake parameters (e.g., turnaround delay and output initiator) are also defined in that section.

A fourth type of handshake method, called software checking, may also be used. Software checking refers to a handshake method that is managed by the applications program. Actually any method, other than a hardwire handshake, could be considered software checking. However, software checking is generally used to refer to a method that can be

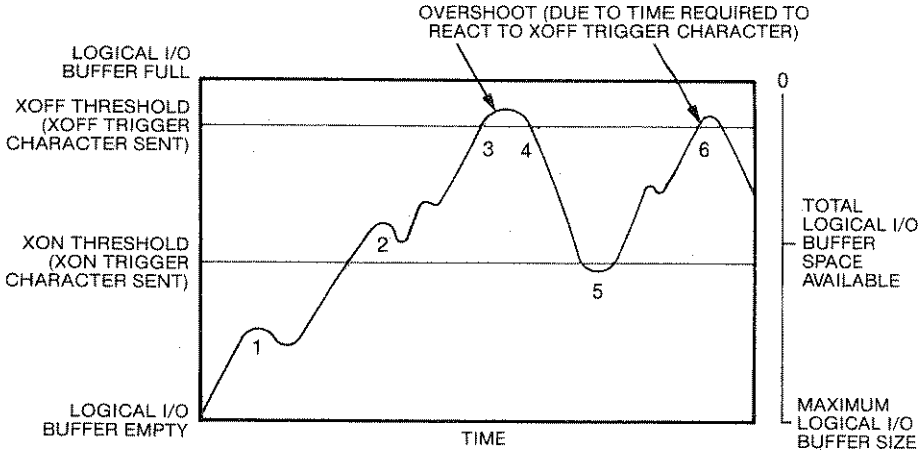
ManualsPlus.com

used on any system, especially if the system cannot implement any of the standard methods mentioned above.

The following paragraphs provide detailed descriptions of the four handshake methods, to help you determine which method to use.

Xon-Xoff Handshake

With the Xon-Xoff handshake method, the plotter controls the data exchange sequence by telling the computer when it has room in its logical I/O buffer for data and when to shut off the flow. The plotter uses buffer threshold indicators (an Xon trigger character and an Xoff trigger character) to prevent buffer overflow.



Xon-Xoff Threshold Levels

As graphics instructions are sent to the plotter by the computer, they are stored in the logical I/O buffer and processed sequentially by the plotter. The preceding figure is representative of the way the Xon-Xoff handshake works; the numbers represent the following:

1. Data enters the buffer faster than it can be acted on by the plotter, and the buffer starts to fill.
2. The plotter begins processing the input data faster than the computer sends it, and the buffer starts to empty.
3. The data enters the buffer at a faster rate than the plotter can process it. The amount of data stored in the buffer reaches the Xoff threshold level, at which point the plotter sends the Xoff trigger character, stopping the flow of data from the computer.
4. Due to a finite delay between the time the plotter sends the Xoff trigger character and the time it takes the computer to react, a slight

ManualsPlus.com

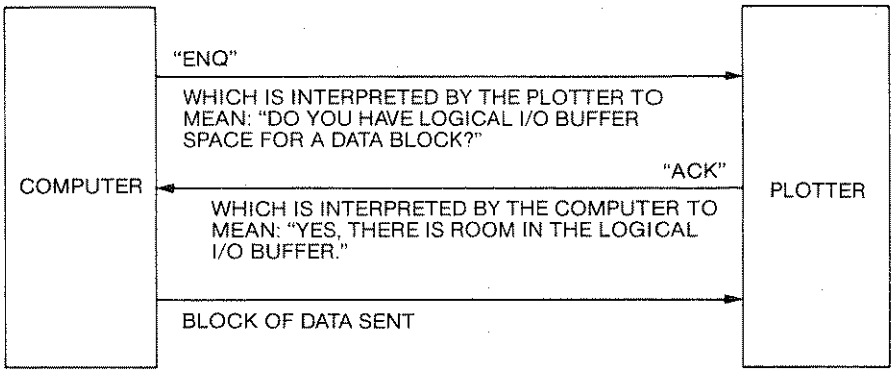
overshoot of the threshold level may occur. In most cases, this overshoot will be much smaller than the default 80 byte block size and results in unused bytes in each block. (Typically, the overshoot is less than 10 characters.)

5. When the amount of stored data drops to the Xon threshold level, the plotter sends the Xon trigger character to signal the computer to resume sending data. The Xon threshold level is automatically set to one block size below the Xoff threshold.
6. Data is again stored in the buffer until all data is transferred or until the Xoff threshold level is exceeded again.

Enquire/Acknowledge Handshake

With the enquire/acknowledge handshake, the computer initiates the data exchange process by querying the plotter about the availability of logical I/O buffer space. The format of the exchange is dependent upon the requirements of the computer.

In its simplest form, the data exchange looks like this:



ENQ/ACK Handshake Protocol, Example 1

In a more complex form, the communication might look like the following example, where the two instructions **ESC**.M250;17;10;13: and **ESC**.H100;5;6: have been sent to specify the variables as:

Turnaround delay = 250 ms

Output trigger character = **DC1** (decimal code 17)

Echo terminate character = **LF** (decimal code 10)

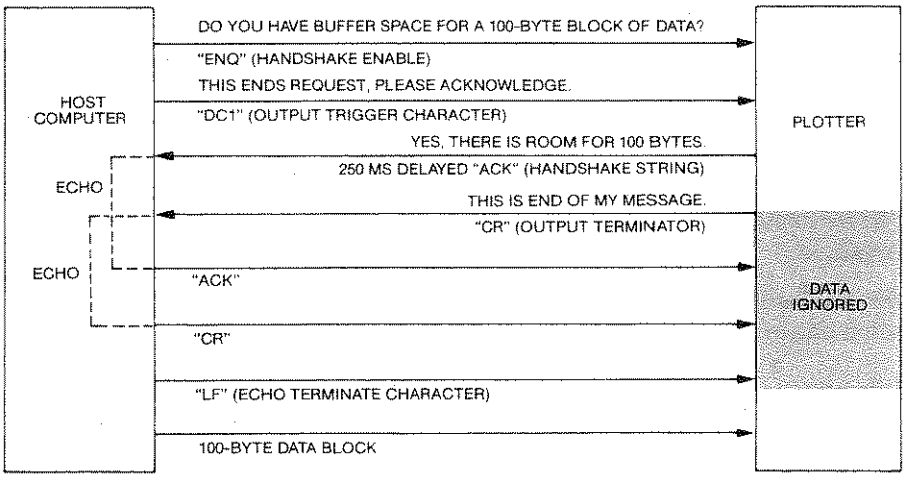
Output terminator = **CR** (decimal code 13)

Data block size = 100 bytes

Enquiry character = **ENQ** (decimal code 5)

Acknowledgment string = **ACK** (decimal code 6)

ManualsPlus.com



ENQ/ACK Handshake Protocol, Example 2

Hardwire Handshake

As the name implies, the hardwire handshake takes place in the hardware rather than the firmware or software. The plotter controls the data exchange sequences by setting the electrical voltage on pin 20 of the connector (CD line) to the computer to signal the computer when to send another block of data. The Data Terminal Ready (DTR or CD) line is set high at power-on and is not set low until available logical I/O buffer space is less than the data block size specified by either ESC.H or ESC.I (default is 80 bytes). DTR remains low until available buffer space is greater than the data block size. By monitoring this line, the computer knows when it can or cannot safely transmit another block of data.

Software Checking Handshake

If you need to develop plotter programs that are independent of a given computer system, you may decide to use instructions in your application program to control the flow of data. This type of handshaking is referred to as software checking. It can be used on almost any computer system that supports input as well as output to a device. This type of handshake method must be used if your system cannot implement any of the other predefined methods.

One method of software checking is a handshake in which a program uses the ESC.B instruction to repeatedly ask the plotter how many bytes of empty space remain in the logical I/O buffer. When the plotter response is bigger than the next block of data, the program will transmit the data block to the plotter. This method may use large amounts of computer and plotter processing time and is inefficient in time-share environments.

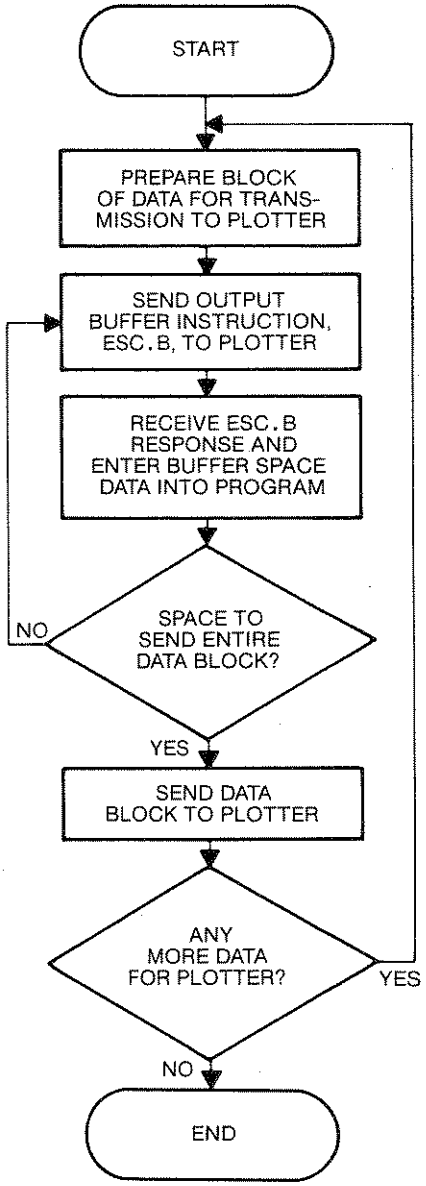
ManualsPlus.com

To match the requirements of the computer system, these variables may be specified for the software checking handshake method by using the appropriate instructions:

- Maximum buffer size (ESC . @ instruction)
- Turnaround delay (ESC . M instruction)
- Output trigger character (ESC . M instruction)
- Echo terminate character (ESC . M instruction)
- Output initiator character (ESC . M instruction)
- Output terminator (ESC . M instruction)
- Intercharacter delay (ESC . N instruction)

The flow diagram on the next page illustrates the functional elements of a typical software checking handshake within a program.

ManualsPlus.com



Software Checking Handshake

Selecting a Predefined Handshake Method from a Program

It is very easy to select a handshake method from a program with the ESC.P instruction. This instruction will set up either an Xon-Xoff, enquire/acknowledge, or hardwire handshake method.

ManualsPlus.com

The ESC.P instruction is designed to set up a handshake method with parameters that are commonly used. However, these parameters may not be the best for your system. For example, if your system uses an enquire/acknowledge handshake but does not use a line feed for the output trigger character, you could use the ESC.M instruction to change the preset value.

If you need to modify any of the predefined values established with ESC.P, refer to the sections on Controlling Plotter Output and on Tailoring Your Own Handshake later in this chapter.

The Set Handshake Mode Instruction, ESC.P

USES: The ESC.P instruction simplifies specification of the handshake method that the plotter uses when communicating with the computer. Use this instruction to select one of three standard handshakes.

SYNTAX: **ESC.P** (<DEC>):

DEFAULT: **ESC.P**: No handshake is set (parameter 0).

EXPLANATION: The ESC.P instruction selects one of three standard handshakes. (Use ESC.@, H, I, M, and N to select a nonstandard handshake.)

The decimal values 0 to 3 are used to specify a handshake method with predefined parameters as follows:

Parameter Value and Handshake Method				Predefined Handshake Parameters Established
0 None	1 Xon-Xoff	2 ENQ/ ACK	3 Hard-wire	
0	50	0	0	Turnaround Delay
0	0	17	0	Output Trigger Character
0	10	10	0	Echo Terminate Character
13	13	13	13	Output Terminator 1
0	0	0	0	Output Terminator 2
0	2	2	0	Handshake Mode
80	80	80	80	Block (Record) Size
1024	1024	1024	1024	Logical I/O Buffer Size
0	0	5	0	Handshake Enable Character
0	17	6	0	Handshake Response
0	10	0	0	Intercharacter Delay
0	19	0	0	Immediate Response String

ManualsPlus.com

Controlling Plotter Output

For a particular application, you may need to slightly modify the format that the plotter uses for outputting data in order to match the requirements of your system. In this case, use the set output mode instruction, ESC.M and the set extended output mode instruction, ESC.N to set certain parameters as explained in the next two sections.

If you need to establish a more specialized type of software checking handshake method, refer to the section on Tailoring Your Own Handshake following the discussions of the ESC.M and ESC.N instructions.

The Set Output Mode Instruction, ESC.M

USES: The ESC.M instruction establishes parameters for the plotter's communication format. Use this instruction to establish a turnaround delay, an output trigger character, an echo terminate character, and an output initiator character. Also, use it to change the output terminator from its default value, carriage return.

SYNTAX: ESC.M [(<DEC>);(<ASC>);(<ASC>);
(<ASC>(;<ASC>));(<ASC>)]:

DEFAULT: ESC.M: Sets the carriage-return character (decimal code 13) as the output terminator. It also specifies that there is no turnaround delay and no output trigger, echo terminate, or output initiator character.

EXPLANATION: A description of the instruction's parameters follows. All parameters are optional.

<DEC> The first parameter specifies the **turnaround delay**. The parameter range is 0 to 9999 milliseconds. If omitted, this parameter assumes its default value of 0 (no turnaround delay).

<ASC> The second parameter specifies a single character which becomes the **output trigger character**. The parameter may be the decimal code of any ASCII character in the range 0 to 126. If omitted, this parameter assumes its default value of 0 (no trigger character).

<ASC> The third parameter specifies a single character which becomes the **echo terminate character**. The parameter may be the decimal code of any ASCII character in the range 0 to 126. If omitted, this parameter assumes its default value of 0 (no echo terminate character).

ManualsPlus.com

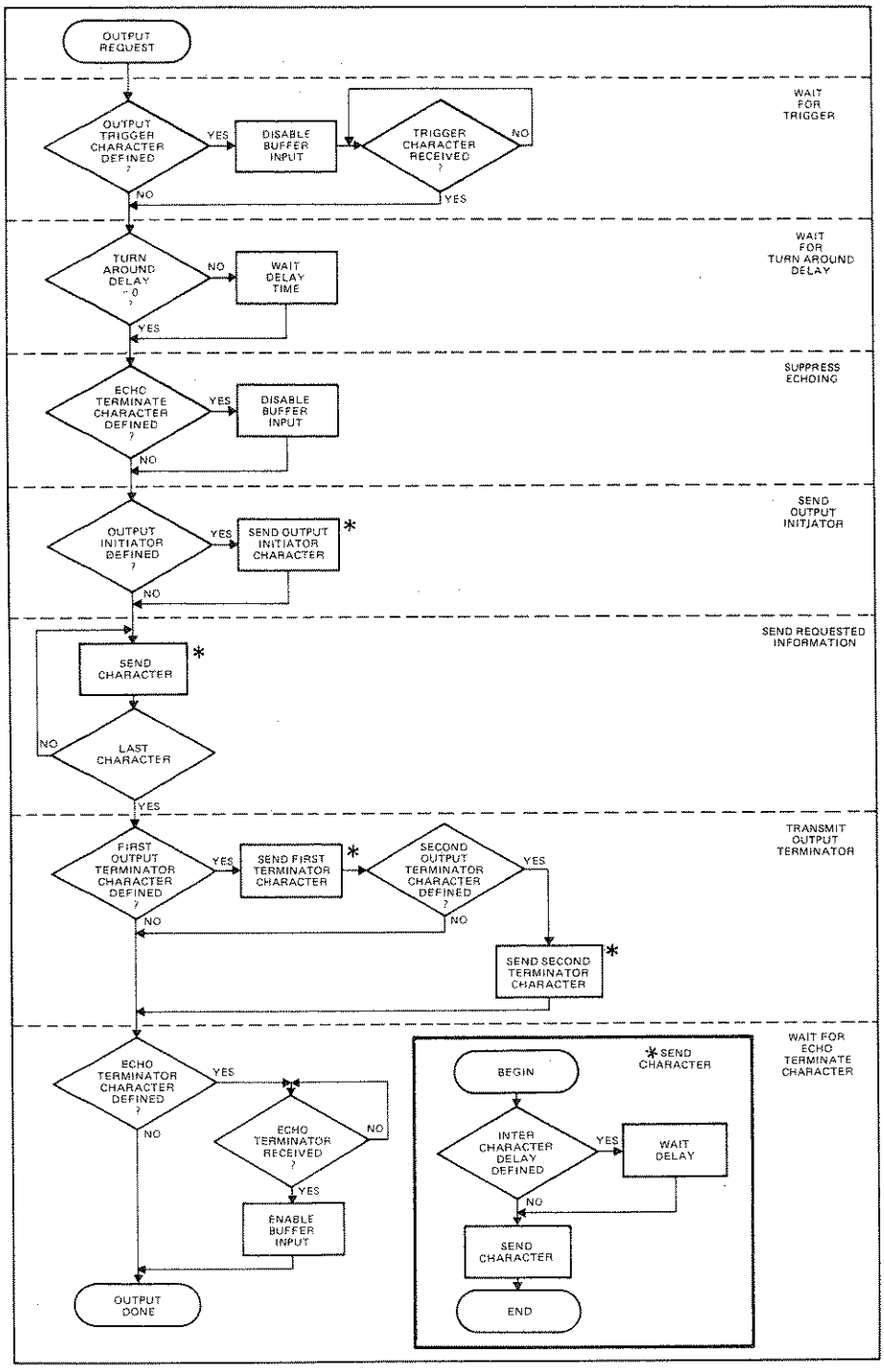
<ASC>...<ASC> The fourth parameter may be the decimal code(s) of one or two ASCII characters in the range 0 to 127. This becomes the **output terminator**. The value 0 is not transmitted, and will terminate the string. If omitted, this parameter defaults to 13, the decimal code of the single ASCII character carriage return (**CR**).

If the fifth parameter is specified, this fourth parameter must consist of two characters, or the second character must be set to 0 or omitted (by entering only a semicolon).

<ASC> The fifth parameter specifies a single character which becomes the **output initiator character**. The parameter may be the decimal code of any ASCII character in the range 0 to 127. If omitted, this parameter assumes its default value 0 (no output initiator character).

The flowchart on the next page depicts plotter output.

ManualsPlus.com



Output Request Flowchart

ManualsPlus.com

The Set Extended Output and Handshake Mode Instruction, ESC.N

USES: The ESC.N instruction establishes parameters for the plotter's communication format. Use this instruction to specify an intercharacter delay for all handshake methods, the immediate response string for the enquire/acknowledge handshake, or the Xoff trigger character(s) for the Xon-Xoff handshake.

SYNTAX: **ESC**.N [(<DEC>);(<ASC>(;...<ASC>))]:

DEFAULT: **ESC**.N: No intercharacter delay and no Xoff trigger character or immediate response string.

EXPLANATION: A description of the instruction's parameters follows. All parameters are optional.

<DEC> The first parameter is the **intercharacter delay**. The parameter range is 0 to 9999 milliseconds.

<ASC>...<ASC> The second parameter is a list of the decimal codes of one to 10 ASCII characters in the range 0 to 127. For an Xon-Xoff handshake, it specifies the **Xoff trigger character(s)**. For an enquire/acknowledge handshake, it specifies the **immediate response string**. Semicolons must separate each parameter in the list.

Tailoring Your Own Handshake

Standard formats for commonly used handshake methods may be selected from the front panel, or under program control with the ESC.P instruction. However, in some applications you may choose to tailor a special handshake method for more efficient operation in your system.

Once you know which handshake method you will use, you can program the plotter to match your system requirements in order to implement the chosen method. This is done by specifying certain variables in device-control instructions which are issued to the plotter at the beginning of each computer session or graphics program. The variables which may be specified through the four handshake methods available to the plotter are:

- Output Trigger Character — The output trigger character, when used, is the last character output by the computer when making a request of a peripheral. Defining this character in an instruction tells the plotter, "Don't respond to my request until you receive this trigger character." This character is often a **DC1** (decimal code 17) or some other nonprinting ASCII character such as **LF** or **CR** or, when using some implementations of BASIC, the **?** (decimal code 63), which does print.

ManualsPlus.com

- Turnaround Delay — The turnaround delay is the length of time the plotter will wait after receiving a computer request before it responds. The purpose of this time delay is to delay the plotter's transmission of requested data until the computer is ready to receive and process it. Systems may require either a turnaround delay or a trigger character, or both.
- Output Initiator Character — The output initiator character is a one-character initiator that is sent by the plotter at the beginning of a string. The output initiator tells the computer, "This starts my transmission." Some computers which require an output initiator expect the start-of-text character **STX** (decimal code 2) as the plotter's output initiator.
- Output Terminator — The output terminator is a one- or two-character terminator that the computer requires the plotter to send at the end of each response to a data request. The output terminator tells the computer, "This completes my transmission." Often, computers expect the carriage-return character **CR** (decimal code 13) as the plotter's output terminator.
- Echo Terminate Character — Echoing is commonly found in full-duplex systems. Use of the echo terminate character in a device-control instruction tells the plotter that the computer will echo all responses and that this echoed data should be ignored (the plotter's logical I/O buffer should be closed) until an echo terminate character is received. When the plotter receives the echo terminate character, it reopens the logical I/O buffer to receive graphics data from the computer. Computers often use the line-feed character **LF** (decimal code 10) as the echo terminator. If the computer does not echo the peripheral's response, this variable must be the **NULL** character (decimal code 0) or must be omitted.
- Intercharacter Delay — Some computers cannot process data as fast as the plotter can transmit it due to limited buffering in their I/O port. This can be compensated for by delaying each transmission from the plotter a period of time as specified by the intercharacter delay variable. This intercharacter delay is added to a turnaround delay (if one has been specified) before the first character is sent by the plotter, and is also inserted before each subsequent character in a string being sent to the computer.
- Enquiry Character — In some systems, the computer sends an enquiry character to ask the plotter if it has room for a block of data, thereby initiating the handshake process. If the Xon-Xoff handshake method is to be established, the **NULL** character (decimal code 0) must be specified as the enquiry character, or the parameter must be omitted. If the enquire/acknowledge handshake method is to be established, an

ManualsPlus.com

ENQ character (decimal code 5) or any other ASCII character besides **NULL** is used.

- Immediate Response String — Certain system environments require an immediate response from the plotter acknowledging the enquiry from the computer. Systems of this type include a computer that transmits data to the plotter after a certain time interval but before receiving a go-ahead signal from the plotter. If the plotter's logical I/O buffer is full and the computer sends more data, the buffer will overflow. The immediate response string prevents this inadvertent transmission of data before the plotter is ready. It is transmitted by the plotter immediately after receipt of an enquiry character and tells the computer, "Wait, I am here and checking my buffer space." The computer will wait indefinitely until it receives an acknowledgment string. Computers frequently require a **DC3** character (decimal code 19) as the immediate response string.
- Acknowledgment String — The acknowledgment string specifies the character or characters that the plotter will send to the computer when the plotter's logical I/O buffer has room for another block of data. Computers frequently require that the **ACK** character (decimal code 6) be used for the acknowledgment string.
- Maximum Logical I/O Buffer Size — This variable establishes the effective usable logical I/O buffer size in the plotter.
- Data Block Size — This is the maximum size of each data block the computer will transmit to the plotter.
- Data Terminal Ready (DTR or CD) Line Control — This variable sets the configuration of the plotter's Data Terminal Ready control line (pin 20) to enable or disable the hardwire handshake method. Pin 20 is held high (+12 V) if hardwire handshake is disabled.
- Xoff Threshold Level — The value of the Xoff threshold level is set with the first parameter of the ESC.I instruction. When the Xon-Xoff handshake method is used, this value is subtracted from the logical I/O buffer size to determine the point at which the plotter sends the Xoff trigger character to the computer, telling it to stop sending data.

  The default Xoff threshold level is set at 80 bytes below the logical I/O buffer size. If the logical I/O buffer size is changed with the ESC.@ instruction, the Xoff threshold will change accordingly.

  During the time it takes the computer to react to an Xoff trigger, a slight overshoot of the threshold level may occur. In most cases, this overshoot will be much smaller than the space allotted by the default 80-byte threshold value, and results in unused bytes in the buffer. (Typically, the overshoot is less than 10 characters.) To maintain maximum throughput, you should specify the smallest possible value that ensures the logical I/O buffer does not overflow.

ManualsPlus.com

- Xoff Trigger Character — This specifies the character string the plotter will use to signal the computer to temporarily stop sending data while the plotter processes what it has already received. The **DC3** character (decimal code 19) is generally used for the Xoff trigger character.
- Xon Trigger Character — This specifies the character string the plotter will use to signal the computer that there is sufficient space in the logical I/O buffer to resume sending data. The **DC1** character (decimal code 17) is generally used for the Xon trigger character.

Quick Reference of Handshake Types

Following is a quick reference and summary of the RS-232-C instructions required for each handshake method. For more detailed information on interfacing and handshake methods and how to choose the best method for your system, refer to the Hewlett-Packard Plotter Note No. 6 (Part No. 5953-4163).

Handled by Operating System			Handled by Software	
Hardwire	**Xon-Xoff**	**ENQ/ACK (Mode 2)**	**ENQ/ACK (Mode 1)**	**Buffer Checking**
ESC.P **or** ESC.@	ESC.P **or** ESC.N and ESC.I	ESC.P **or** ESC.I	— ESC.M and ESC.H	— ESC.M and ESC.B

Device-Control Instructions and Values Used to Establish Handshake Methods

The following instructions may be used to establish any handshake method. The ESC.@ and ESC.B instructions are presented in Chapter 14; the other instructions are presented in this chapter.

ESC.@ logical I/O buffer size; set configuration options (which include hardwire handshake):

ESC.B (No parameters — This instruction outputs the number of bytes currently available for data in the logical I/O buffer. Response is an integer of five digits or less with no decimal point.)

ESC.H block size; enquiry character; acknowledgment string:

ESC.I block size (in ENQ/ACK handshake) or Xoff threshold level (in Xon-Xoff handshake); enquiry character or omitted; acknowledgment character(s) or Xon trigger character(s):

ManualsPlus.com

ESC.M turnaround delay; output trigger character; echo terminate character; output terminator(s); output initiator:

ESC.N intercharacter delay; immediate response string or Xoff trigger characters:

ESC.P handshake method:

A Summary of RS-232-C Device-Control Instructions

The RS-232-C interface instructions are a special type of device-control instructions. The RS-232-C instructions discussed in this chapter are used to establish and control serial communications between the plotter and a computer. A discussion of general-purpose device-control instructions, including the use of monitor mode for program debugging and how to interpret error conditions, appears in Chapter 14. Refer also to Chapter 14 for a description of syntax conventions for device-control instructions, which are different from HP-GL syntax conventions. Remember that all device-control instructions are immediately executed when received. Unlike HP-GL instructions, they are not stored in the I/O buffer.

Serial communications means that bytes of information are transmitted in a sequential, rather than parallel, format. The plotter goes into serial mode if the first data it receives enters through a serial port (either **COMPUTER/MODEM** or **TERMINAL**).

NOTE: The ESC.H and ESC.I instructions are mutually exclusive. Depending on the requirements of the computer system, either instruction can be used for the enquire/acknowledge handshake, but only ESC.I can be used for the Xon-Xoff handshake. ■

The Set Handshake Mode 1 Instruction, ESC.H

USES: The ESC.H instruction is useful for setting up a handshake method where a program controls the handshaking. It may be used with the enquire/acknowledge handshake to establish the plotter's implementation of the handshake protocol. Use this instruction to configure the plotter for enquire/acknowledge handshake when the computer requires that the plotter's output is sent in accordance with the parameters set in the ESC.M and ESC.N instructions.

SYNTAX: **ESC.H** [(<DEC>);(<ASC>);(<ASC>(;...<ASC>))]:

DEFAULT: **ESC.H**: The enquire/acknowledge handshake is disabled. Data block size is 80 bytes, and there is no enquiry character or acknowledgment string. If, however, the computer is configured to send an **ENQ**

ManualsPlus.com

anytime it is ready to send data to the plotter, the plotter will automatically respond with **ACK** when it receives **ENQ**. This "dummy handshake" is not dependent upon available buffer space and does not protect against logical I/O buffer overflow.

EXPLANATION: A description of the parameters is given next. Remember that this instruction may be used for an enquire/acknowledge handshake, but it may *not* be used for an Xon-Xoff handshake.

<DEC> The first parameter specifies the **data block size**. Parameter range is 0 to 32767. The data block size must be set to less than the current logical I/O buffer size. Default block size set when the parameter is omitted is 80 bytes.

<ASC> This parameter sets the **enquiry character**. The parameter may be the decimal code of any ASCII character in the range 0 to 126. If the parameter is omitted, it assumes the default value 0 (**NULL** character) disabling enquire/acknowledge handshake. Any value other than 0 enables enquire/acknowledge handshake. However, the value 5 (enquiry character, **ENQ**) is generally used.

<ASC>...<ASC> This is a list of one to 10 characters, separated by semicolons, which specifies the **acknowledgment string**. Decimal codes of ASCII characters in the range 0 to 127 are valid. The value 0 is not transmitted and will terminate the string. The value 6 (acknowledgment character, **ACK**) is generally used. If the parameter is omitted, it assumes the default value 0 and no characters are sent.

The ESC.M and ESC.N parameters that affect output responses of handshake mode 1, handshake mode 2, and all output instructions are shown in the table on the following page. Use handshake mode 1, set by ESC.H, or handshake mode 2, set by ESC.I, depending on the requirements of the computer operating system.

ManualsPlus.com

Parameter Inclusion in Plotter Responses

ESC.M/ESC.N Parameters	ESC.H Handshake Mode 1	ESC.I Handshake Mode 2		Output Instructions
	ENQ/ACK	ENQ/ACK	Xon-Xoff	
Turnaround Delay	Yes	Yes	No	Yes
Output Trigger Character	Yes	No	No	Yes
Echo Terminator	Yes	No	No	Yes
Output Terminator	Yes	No	No	Yes
Output Initiator*	No	No	No	Yes
Intercharacter Delay	Yes	Yes	Yes	Yes
Immediate Response String	Yes	Yes	No	No

*If an output initiator is required on enquiry responses, it should be specified as the first character of the acknowledgment string and/or the immediate response string, depending on the system.

Example — Using ESC.H with an Enquire/Acknowledge Handshake

ESC. H 132;19;20;7: will set the block size to 132 bytes, the ASCII character **DC3** as the enquiry character, and the two characters **DC4** and **BEL** as the acknowledgment string. Since ESC.H sets handshake mode 1, the currently defined output terminator, output trigger character, and echo terminator as well as the turnaround delay, intercharacter delay, and immediate response string, are used when the response string, **DC4 BEL** is sent.

The Set Handshake Mode 2 Instruction, ESC.I

USES: The ESC.I instruction can be used with a hardwire handshake to change the data block size threshold for the DTR line. ESC.I is also used in large computer systems where handshaking is handled by the operating system via input/output (I/O) driver software. With the enquire/acknowledge handshake, it establishes the data block size, the enquiry character, and the acknowledgment string when the computer does not expect the parameters set by ESC.M to be included in the response to the enquiry character. With the Xon-Xoff handshake, it sets the Xoff threshold level and the Xon trigger character.

ManualsPlus.com

SYNTAX: **ESC**.I [(<DEC>);(<ASC>);(<ASC>(;...<ASC>))]:

DEFAULT: **ESC**.I: Neither Xon-Xoff nor enquire/acknowledge handshake is enabled. Data block size is 80 bytes, and there is no enquiry character or acknowledgment string. If, however, the computer is configured to send an **ENQ** anytime it is ready to send data to the plotter, the plotter will automatically respond with **ACK** when it receives **ENQ**. This "dummy handshake" is not dependent upon available buffer space and does not protect against logical I/O buffer overflow.

EXPLANATION: With handshake mode 2, some of the parameters set by the ESC.M instruction do not effect when the plotter outputs the acknowledgment string or the Xon and Xoff trigger characters. Refer to the table under the ESC.H instruction to see which parameters are used. This information will help you choose whether to use ESC.I or ESC.H.

A description of the parameters, as interpreted for the Xon-Xoff handshake, is given next. Refer to ESC.H for a description of the parameters as interpreted for the enquire/acknowledge handshake.

<DEC> The first parameter sets the **Xoff threshold level** by specifying a value that is subtracted from the logical I/O buffer size to determine the point at which the Xoff trigger character is to be sent.

The Xon threshold level is set at one Xoff-threshold value below the Xoff threshold level. The Xoff threshold level cannot be set at less than one-half the logical I/O buffer size. If you specify a threshold value that would cause the threshold level to be less than one-half the logical I/O buffer size, the Xoff threshold level will automatically be set at one-half the logical I/O buffer size, and the Xon threshold level will be set to zero.

<ASC> To specify the Xon-Xoff handshake method, omit this parameter by entering only a semicolon or the value 0 followed by the semicolon. To enable Xon-Xoff handshake, the next parameter, which specifies an Xon trigger character(s), must also be included.

<ASC>...<ASC> One to 10 characters, separated by semicolons, which specify the **Xon trigger character(s)**. Decimal codes of ASCII characters in the range 0 to 127 are valid. The value 0 is not transmitted and will terminate the string.

Example — Using ESC.I with a Hardwire Handshake

ESC.I10: will set the data block size threshold for the DTR line to 10 bytes. Since the other parameters of ESC.I are unnecessary for a hardwire handshake, they are not included.

ManualsPlus.com

Example — Using ESC.I with an Enquire/Acknowledge Handshake

ESC.I;5;6: will set the block size to its default value of 80 bytes, the ASCII character **ENQ** as the enquiry character, and the ASCII character **ACK** as the acknowledgment string. Only the turnaround delay (set with ESC.M), intercharacter delay, and immediate response string (set with ESC.N), if any, are used when sending the acknowledgment string. No output initiator will precede it, even if one is defined, and no output terminator will follow it.

Example — Using ESC.I with an Xon-Xoff Handshake

ESC.I81;;17: will set the Xoff threshold level to 81 bytes less than the current logical I/O buffer size (the Xoff character will be sent when 81 empty bytes remain in the plotter's logical I/O buffer) and the Xon trigger character to **DC1**. The second parameter is defaulted as required for this handshake.

NOTE: The Xoff trigger character must be specified with the ESC.N instruction. ■

The Abort Device Control Instruction, ESC.J

USES: The ESC.J instruction aborts the execution of any device-control instruction. Use this instruction in an initialization sequence when you first access the plotter.

SYNTAX: **ESC**.J

EXPLANATION: This instruction aborts any device-control instruction that may be partially parsed or executed. The remaining parameters of partially parsed instructions are defaulted. All pending or partially transmitted output requests, from either HP-GL or device-control instructions, are immediately terminated, including handshake outputs. Intermediate output operations, such as turnaround delay and echo suppression, are aborted and buffer input is enabled. The handshake and output mode parameters remain as specified. There is no response from the plotter.

The Plotter-On Instruction, ESC.Y or ESC.(

USES: The ESC.Y or ESC.(instruction enables the plotter to accept data and interpret it as HP-GL or device-control instructions. Use this instruction to establish the programmed-on state in either the remote or local mode.

ManualsPlus.com

SYNTAX: **ESC.** Y
or
ESC. (

EXPLANATION: Depending on the selected operating mode, the plotter scans for a plotter-on instruction from either the computer or the terminal. After receipt of this instruction from the selected source, the plotter is programmed-on and will interpret all subsequent data from the selected source as plotter instructions. If the plotter is already programmed-on, it will ignore this instruction.

The Plotter-Off Instruction, ESC.Z or ESC.)

USES The ESC.Z or ESC.) instruction disables the plotter so that it will only accept a plotter-on instruction. Use this instruction to establish the programmed-off state in either the remote or local mode.

SYNTAX: **ESC.** Z
or
ESC.)

EXPLANATION: After a plotter-off instruction is received, the plotter is programmed-off. Any HP-GL instructions remaining in the I/O buffer are executed, but no additional instructions will be accepted until a plotter-on instruction is received. If the plotter is already programmed-off, it will ignore this instruction.

In an eavesdrop configuration, a break signal from the terminal will have the same effect as a plotter-off instruction. In a stand-alone configuration, the plotter cannot be programmed-off.

ManualsPlus.com

Appendix A
Reference Material

Binary Coding and Conversions

Binary is a base 2 number system using only 1's and 0's. By giving the 1's and 0's positional value, any decimal number can be represented. For example, this diagram shows how decimal 41 = binary 101001:

Decimal

$$(4\times 10^1)+(1\times 10^0)$$
$$(4\times 10)+(1\times 1)$$
$$4 \qquad 1_{10}$$

Binary

$$(1\times 2^5)+(0\times 2^4)+(1\times 2^3)+(0\times 2^2)+(0\times 2^1)+(1\times 2^0)$$
$$(1\times 32)+(0\times 16)+(1\times 8)+(0\times 4)+(0\times 2)+(1\times 1)$$
$$1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1_2$$

Binary-Decimal Conversions

To convert from binary to decimal, add the positional values of the 1's. From the above example, this would be:

$$2^5+2^3+2=32+8+1=41$$

To convert from decimal to binary, divide the decimal number by 2. The remainder is the binary equivalent. For example:

		Remainder (read up)
2 ⟌41	→	1
2 ⟌20	→	0
2 ⟌10	→	0
2 ⟌5	→	1
2 ⟌2	→	0
2 ⟌1	→	1

= Binary 101001

ManualsPlus.com

ASCII Character Codes

Numbers are often used as a code to represent not only values, but also alphanumeric characters such as "A" or "," or "x" or "2". One of the most common computer codes used is ASCII*. ASCII is an eight-bit code, containing seven data bits and one parity bit. The plotter uses ASCII for most I/O operations. No parity bit is used. For example:

Character	ASCII Binary Code	ASCII Decimal Code
A	01000001	65
B	01000010	66
?	00111111	63

Character Sets and ASCII Codes

The HP 7550 has 20 character sets, each available in two fonts. They are listed in the following table.

Character Set No.		Description	ISO Registration Number
Fixed-Space	Variable-Space		
0	10	ANSI ASCII	006
1	11	HP 9825 HPL Character Set	—
2	12	French/German	—
3	13	Scandinavian	—
4	14	Spanish/Latin American	—
5	15	Special Symbols	—
6	16	JIS ASCII	014
7	17	Roman Extensions	—
8	18	Katakana	013
9	19	ISO IRV (International Reference Version)	002
30	40	ISO Swedish	010
31	41	ISO Swedish for Names	011
32	42	ISO Norwegian, Version 1	060
33	43	ISO German	021

*American Standard Code for Information Interchange

ManualsPlus.com

Character Set No.		Description	ISO Registration Number
Fixed-Space	Variable-Space		
34	44	ISO French	025
35	45	ISO British	004
36	46	ISO Italian	015
37	47	ISO Spanish	017
38	48	ISO Portuguese	016
39	49	ISO Norwegian, Version 2	061

The plotter's reactions to nonprinting ASCII control characters (decimal codes 0–32, 127, and 142–143) are shown in the following table. These reactions are true regardless of the character set being used for labeling.

Reaction to Nonprinting Control Characters

Decimal Value	ASCII Character	All Sets
0	NULL	No Operation (NOP)
1	SOH	NOP
2	STX	NOP
3	ETX	Terminate Label Instruction
4	ETO	NOP
5	ENQ	NOP
6	ACK	NOP
7	BEL	NOP
8	BS	Backspace
*9	HT	Horizontal Tab (½ Backspace)
10	LF	Line Feed
11	VT	Inverse Line Feed
12	FF	NOP
13	CR	Carriage Return
14	SO	Shift-Out (Select Alternate Character Set)

*Using control character horizontal tab (decimal code 9) inside a label string moves the pen one-half character space back (equivalent to a CP −5, 0). Use this tab with character set 8, Katakana, where spacing between symbols can alter the meaning of the symbol and hence the word or phrase.

ManualsPlus.com

Reaction to Nonprinting Control Characters (Continued)

Decimal Value	ASCII Character	All Sets
15	SI	Shift-In (Select Standard Character Set)
16	DLE	NOP
17	DC1	NOP
18	DC2	NOP
19	DC3	NOP
20	DC4	NOP
21	NAK	NOP
22	SYN	NOP
23	ETB	NOP
24	CAN	NOP
25	EM	NOP
26	SUB	NOP
27	ESC	NOP
28	FS	NOP
29	GS	NOP
30	RS	NOP
31	US	NOP
32	SP	Space
127	DEL	NOP
142	SS2	Single Shift to Slot G2 (ISO 8-Bit Mode Only)
143	SS3	Single Shift to Slot G3 (ISO 8-Bit Mode Only)

The plotter's character sets are shown on the following pages. All printing ASCII characters and their decimal codes (33–126) are listed for each font in each character set. The fixed-space fonts are shown first, followed by the variable-space fonts.

NOTE: Each of the shaded symbols is automatically backspaced one character before it is drawn. Therefore, when an accented letter is required, the letter should be entered first, followed by the backspaced character. In addition, the special centered symbols in character sets 5 and 15 (ASCII decimal codes 65–81) are designed for use in symbol mode with the SM instruction. When used in a label instruction, spacing will be irregular and may produce undesirable results. ■

ManualsPlus.com

Printing Characters, Fixed-Space Fonts

DECIMAL VALUE	SET																			
	0	1	2	3	4	5	6	7	8	9	30	31	32	33	34	35	36	37	38	39
33	!	!	!	!	!	!	!	À	｡	!	!	!	!	!	!	!	!	!	!	!
34	"	"	"	"	"	"	"	Â	｢	"	"	"	"	"	"	"	"	"	"	"
35	#	#	£	£	¿	#	#	È	｣	#	#	#	#	#	£	£	£	£	#	§
36	$	$	$	$	$	$	$	Ê	､	¤	¤	¤	$	$	$	$	$	$	$	$
37	%	%	%	%	%	%	%	Ë	･	%	%	%	%	%	%	%	%	%	%	%
38	&	&	&	&	&	&	&	Î	ｦ	&	&	&	&	&	&	&	&	&	&	&
39	'	'	'	'	'	'	'	Ï	ｧ	'	'	'	'	'	'	'	'	'	'	'
40	(	(	(	(	(	(	(	´	ｨ	(	(	(	(	(	(	(	(	(	(	(
41	)	)	)	)	)	)	)	`	ｩ	)	)	)	)	)	)	)	)	)	)	)
42	*	*	*	*	*	*	*	ˆ	ｪ	*	*	*	*	*	*	*	*	*	*	*
43	+	+	+	+	+	+	+	¨	ｫ	+	+	+	+	+	+	+	+	+	+	+
44	,	,	,	,	,	,	,	~	ｬ	,	,	,	,	,	,	,	,	,	,	,
45	-	-	-	-	-	-	-	Ù	ｭ	-	-	-	-	-	-	-	-	-	-	-
46	.	.	.	.	.	.	.	Û	ｮ	.	.	.	.	.	.	.	.	.	.	.
47	/	/	/	/	/	/	/	£	ｯ	/	/	/	/	/	/	/	/	/	/	/
48	0	0	0	0	0	0	0	¯	ｰ	0	0	0	0	0	0	0	0	0	0	0
49	1	1	1	1	1	1	1		ｱ	1	1	1	1	1	1	1	1	1	1	1
50	2	2	2	2	2	2	2		ｲ	2	2	2	2	2	2	2	2	2	2	2
51	3	3	3	3	3	3	3	°	ｳ	3	3	3	3	3	3	3	3	3	3	3
52	4	4	4	4	4	4	4	Ç	ｴ	4	4	4	4	4	4	4	4	4	4	4
53	5	5	5	5	5	5	5	ç	ｵ	5	5	5	5	5	5	5	5	5	5	5
54	6	6	6	6	6	6	6	Ñ	ｶ	6	6	6	6	6	6	6	6	6	6	6
55	7	7	7	7	7	7	7	ñ	ｷ	7	7	7	7	7	7	7	7	7	7	7
56	8	8	8	8	8	8	8	¡	ｸ	8	8	8	8	8	8	8	8	8	8	8
57	9	9	9	9	9	9	9	¿	ｹ	9	9	9	9	9	9	9	9	9	9	9
58	:	:	:	:	:	:	:	¤	ｺ	:	:	:	:	:	:	:	:	:	:	:
59	;	;	;	;	;	:	;	£	ｻ	;	;	;	;	;	;	;	;	;	;	;
60	<	<	<	<	<	<	<	¥	ｼ	<	<	<	<	<	<	<	<	<	<	<
61	=	=	=	=	=	=	=	§	ｽ	=	=	=	=	=	=	=	=	=	=	=
62	>	>	>	>	>	>	>	ƒ	ｾ	>	>	>	>	>	>	>	>	>	>	>
63	?	?	?	?	?	?	?	¢	ｿ	?	?	?	?	?	?	?	?	?	?	?
64	@	@	@	@	@	@	@	â	ﾀ	@	@	É	@	§	à	@	§	§	§	@

ManualsPlus.com

Printing Characters, Fixed-Space Fonts (Continued)

DECIMAL VALUE	SET 0	1	2	3	4	5	6	7	8	9	30	31	32	33	34	35	36	37	38	39
65	A	A	A	A	A	□	A	ê	∓	A	A	A	A	A	A	A	A	A	A	A
66	B	B	B	B	B	◇	B	ô	ッ	B	B	B	B	B	B	B	B	B	B	B
67	C	C	C	C	C	△	C	û	∓	C	C	C	C	C	C	C	C	C	C	C
68	D	D	D	D	D	+	D	á	ト	D	D	D	D	D	D	D	D	D	D	D
69	E	E	E	E	E	×	E	é	ナ	E	E	E	E	E	E	E	E	E	E	E
70	F	F	F	F	F	◇	F	ó	二	F	F	F	F	F	F	F	F	F	F	F
71	G	G	G	G	G	↑	G	ú	ヌ	G	G	G	G	G	G	G	G	G	G	G
72	H	H	H	H	H	⊠	H	à	ネ	H	H	H	H	H	H	H	H	H	H	H
73	I	I	I	I	I	≥	I	è	ノ	I	I	I	I	I	I	I	I	I	I	I
74	J	J	J	J	J	↘	J	ò	ハ	J	J	J	J	J	J	J	J	J	J	J
75	K	K	K	K	K	⊠	K	ù	ヒ	K	K	K	K	K	K	K	K	K	K	K
76	L	L	L	L	L	✳	L	ä	フ	L	L	L	L	L	L	L	L	L	L	L
77	M	M	M	M	M	⊠	M	ë	ヘ	M	M	M	M	M	M	M	M	M	M	M
78	N	N	N	N	N	↓	N	ö	ホ	N	N	N	N	N	N	N	N	N	N	N
79	O	O	O	O	O	✿	O	ü	マ	O	O	O	O	O	O	O	O	O	O	O
80	P	P	P	P	P	_	P	Å	Ξ	P	P	P	P	P	P	P	P	P	P	P
81	Q	Q	Q	Q	Q	↓	Q	î	ム	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
82	R	R	R	R	R	R	R	Ø	メ	R	R	R	R	R	R	R	R	R	R	R
83	S	S	S	S	S	S	S	Æ	モ	S	S	S	S	S	S	S	S	S	S	S
84	T	T	T	T	T	T	T	å	ヤ	T	T	T	T	T	T	T	T	T	T	T
85	U	U	U	U	U	U	U	í	ユ	U	U	U	U	U	U	U	U	U	U	U
86	V	V	V	V	V	V	V	ø	ヨ	V	V	V	V	V	V	V	V	V	V	V
87	W	W	W	W	W	W	W	æ	ラ	W	W	W	W	W	W	W	W	W	W	W
88	X	X	X	X	X	X	X	Ä	リ	X	X	X	X	X	X	X	X	X	X	X
89	Y	Y	Y	Y	Y	Y	Y	ì	ル	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
90	Z	Z	Z	Z	Z	Z	Z	Ö	レ	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
91	[	[	[	Ø	[	[	[	Ü	ロ	[	Ä	Ä	Æ	Ä	°	[	°	¡	Ã	Æ
92	\	√	ç	Æ	¡	\	¥	É	ワ	\	Ö	Ö	Ø	Ö	ç	\	ç	Ñ	Ç	Ø
93	]	]	]	ø	]	]	]	ï	ン	]	Å	Å	Å	Ü	§	]	é	¿	Õ	Å
94	^	↑	^	æ	^	^	^	ß	"	^	^	Ü	^	^	^	^	^	^	^	^
95	_	_	_	_	_	_	_	Ô	°	_	_	_	_	_	_	_	_	_	_	_
96	`	`	`	`	`	`	`	Á		`	`	é	`	`	`	`	ù	`	`	`

ManualsPlus.com

Printing Characters, Fixed-Space Fonts (Continued)

DECIMAL VALUE	SET 0	1	2	3	4	5	6	7	8	9	30	31	32	33	34	35	36	37	38	39
97	a	a	a	a	a	∩	a	Ã		a	a	a	a	a	a	a	a	a	a	a
98	b	b	b	b	b	⊃	b	ã		b	b	b	b	b	b	b	b	b	b	b
99	c	c	c	c	c	⊂	c	Đ		c	c	c	c	c	c	c	c	c	c	c
100	d	d	d	d	d	∪	d	đ		d	d	d	d	d	d	d	d	d	d	d
101	e	e	e	e	e	⊆	e	Í		e	e	e	e	e	e	e	e	e	e	e
102	f	f	f	f	f	≡	f	Ì		f	f	f	f	f	f	f	f	f	f	f
103	g	g	g	g	g	≅	g	Ó		g	g	g	g	g	g	g	g	g	g	g
104	h	h	h	h	h	≈	h	Ò		h	h	h	h	h	h	h	h	h	h	h
105	i	i	i	i	i	~	i	Õ		i	i	i	i	i	i	i	i	i	i	i
106	j	j	j	j	j	≤	j	õ		j	j	j	j	j	j	j	j	j	j	j
107	k	k	k	k	k	≥	k	Š		k	k	k	k	k	k	k	k	k	k	k
108	l	l	l	l	l	≠	l	š		l	l	l	l	l	l	l	l	l	l	l
109	m	m	m	m	m	Δ	m	Ú		m	m	m	m	m	m	m	m	m	m	m
110	n	n	n	n	n	Π	n	Ÿ		n	n	n	n	n	n	n	n	n	n	n
111	o	o	o	o	o	Σ	o	ÿ		o	o	o	o	o	o	o	o	o	o	o
112	p	p	p	p	p	±	p	Þ		p	p	p	p	p	p	p	p	p	p	p
113	q	q	q	q	q	∓	q	þ		q	q	q	q	q	q	q	q	q	q	q
114	r	r	r	r	r	→	r			r	r	r	r	r	r	r	r	r	r	r
115	s	s	s	s	s	↑	s			s	s	s	s	s	s	s	s	s	s	s
116	t	t	t	t	t	←	t			t	t	t	t	t	t	t	t	t	t	t
117	u	u	u	u	u	↓	u			u	u	u	u	u	u	u	u	u	u	u
118	v	v	v	v	v	∫	v	–		v	v	v	v	v	v	v	v	v	v	v
119	w	w	w	w	w	÷	w	¼		w	w	w	w	w	w	w	w	w	w	w
120	x	x	x	x	x	*	x	½		x	x	x	x	x	x	x	x	x	x	x
121	y	y	y	y	y	∇	y	ª		y	y	y	y	y	y	y	y	y	y	y
122	z	z	z	z	z	°	z	º		z	z	z	z	z	z	z	z	z	z	z
123	{	¤	¨	~	~	{	{	≪		{	ä	ä	æ	ä	é	{	à	°	ã	æ
124	|	⊢	°	¨	~	|	|	□		ö	ö	ø	ö	ù	|	ò	ñ	ç	ø
125	}	⊣	~	¨	~	}	}	≫		}	å	å	å	ü	è	}	è	ç	õ	å
126	~	~	¨	¨	~	~	~	±		~	~	ü	~	ß	^	~	ì	~	°	~

ManualsPlus.com

Printing Characters, Variable-Space Fonts

DECIMAL VALUE	SET 10	11	12	13	14	15	16	17	18	19	40	41	42	43	44	45	46	47	48	49
33	!	!	!	!	!	!	!	À	。	!	!	!	!	･	!	!	!	!	!	!
34	"	"	"	"	"	"	"	Â	「	"	"	"	"	"	"	"	"	"	"	"
35	#	#	£	£	¿	#	#	È	」	#	#	#	#	#	£	£	£	£	#	§
36	$	$	$	$	$	$	$	Ê	、	¤	¤	¤	$	$	$	$	$	$	$	$
37	%	%	%	%	%	%	%	Ë	・	%	%	%	%	%	%	%	%	%	%	%
38	&	&	&	&	&	&	&	Î	ヲ	&	&	&	&	&	&	&	&	&	&	&
39	'	'	'	'	'	'	'	Ï	ァ	'	'	'	'	'	'	'	'	'	'	'
40	(	(	(	(	(	(	(	´	ィ	(	(	(	(	(	(	(	(	(	(	(
41	)	)	)	)	)	)	)	`	ゥ	)	)	)	)	)	)	)	)	)	)	)
42	*	*	*	*	*	*	*	ˆ	ェ	*	*	*	*	*	*	*	*	*	*	*
43	+	+	+	+	+	+	+	¨	ォ	+	+	+	+	+	+	+	+	+	+	+
44	,	,	,	,	,	,	,	~	ャ	,	,	,	,	,	,	,	,	,	,	,
45	−	−	−	−	−	−	−	Ù	ュ	−	−	−	−	−	−	−	−	−	−	−
46	.	.	.	.	.	.	.	Û	ョ	.	.	.	.	.	.	.	.	.	.	.
47	/	/	/	/	/	/	/	£	ッ	/	/	/	/	/	/	/	/	/	/	/
48	0	0	0	0	0	0	0	¯	ー	0	0	0	0	0	0	0	0	0	0	0
49	1	1	1	1	1	1	1		ア	1	1	1	1	1	1	1	1	1	1	1
50	2	2	2	2	2	2	2		イ	2	2	2	2	2	2	2	2	2	2	2
51	3	3	3	3	3	3	3	¸	ウ	3	3	3	3	3	3	3	3	3	3	3
52	4	4	4	4	4	4	4	Ç	エ	4	4	4	4	4	4	4	4	4	4	4
53	5	5	5	5	5	5	5	ç	オ	5	5	5	5	5	5	5	5	5	5	5
54	6	6	6	6	6	6	6	Ñ	カ	6	6	6	6	6	6	6	6	6	6	6
55	7	7	7	7	7	7	7	ñ	キ	7	7	7	7	7	7	7	7	7	7	7
56	8	8	8	8	8	8	8	¡	ク	8	8	8	8	8	8	8	8	8	8	8
57	9	9	9	9	9	9	9	¿	ケ	9	9	9	9	9	9	9	9	9	9	9
58	:	:	:	:	:	:	:	¤	コ	:	:	:	:	:	:	:	:	:	:	:
59	;	;	;	;	;	;	;	£	サ	;	;	;	;	;	;	;	;	;	;	;
60	<	<	<	<	<	<	<	¥	シ	<	<	<	<	<	<	<	<	<	<	<
61	=	=	=	=	=	=	=	§	ス	=	=	=	=	=	=	=	=	=	=	=
62	>	>	>	>	>	>	>	ƒ	セ	>	>	>	>	>	>	>	>	>	>	>
63	?	?	?	?	?	?	?	¢	ソ	?	?	?	?	?	?	?	?	?	?	?
64	@	@	@	@	@	@	@	â	タ	@	@	É	@	§	à	@	§	§	§	@

ManualsPlus.com

Printing Characters, Variable-Space Fonts (Continued)

DECIMAL VALUE	SET 10	11	12	13	14	15	16	17	18	19	40	41	42	43	44	45	46	47	48	49
65	A	A	A	A	A		A	ê	ヲ	A	A	A	A	A	A	A	A	A	A	A
66	B	B	B	B	B		B	ô	ッ	B	B	B	B	B	B	B	B	B	B	B
67	C	C	C	C	C		C	û	ヲ	C	C	C	C	C	C	C	C	C	C	C
68	D	D	D	D	D		D	á	ト	D	D	D	D	D	D	D	D	D	D	D
69	E	E	E	E	E		E	é	ナ	E	E	E	E	E	E	E	E	E	E	E
70	F	F	F	F	F		F	ó	ニ	F	F	F	F	F	F	F	F	F	F	F
71	G	G	G	G	G		G	ú	ヌ	G	G	G	G	G	G	G	G	G	G	G
72	H	H	H	H	H		H	à	ネ	H	H	H	H	H	H	H	H	H	H	H
73	I	I	I	I	I		I	è	ノ	I	I	I	I	I	I	I	I	I	I	I
74	J	J	J	J	J		J	ò	ハ	J	J	J	J	J	J	J	J	J	J	J
75	K	K	K	K	K		K	ù	ヒ	K	K	K	K	K	K	K	K	K	K	K
76	L	L	L	L	L		L	ä	フ	L	L	L	L	L	L	L	L	L	L	L
77	M	M	M	M	M		M	ë	ヘ	M	M	M	M	M	M	M	M	M	M	M
78	N	N	N	N	N		N	ö	ホ	N	N	N	N	N	N	N	N	N	N	N
79	O	O	O	O	O		O	ü	マ	O	O	O	O	O	O	O	O	O	O	O
80	P	P	P	P	P	_	P	Å	ミ	P	P	P	P	P	P	P	P	P	P	P
81	Q	Q	Q	Q	Q	‹	Q	î	ム	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
82	R	R	R	R	R	R	R	Ø	メ	R	R	R	R	R	R	R	R	R	R	R
83	S	S	S	S	S	S	S	Æ	モ	S	S	S	S	S	S	S	S	S	S	S
84	T	T	T	T	T	T	T	å	ヤ	T	T	T	T	T	T	T	T	T	T	T
85	U	U	U	U	U	U	U	í	ユ	U	U	U	U	U	U	U	U	U	U	U
86	V	V	V	V	V	V	V	ø	ヨ	V	V	V	V	V	V	V	V	V	V	V
87	W	W	W	W	W	W	W	æ	ラ	W	W	W	W	W	W	W	W	W	W	W
88	X	X	X	X	X	X	X	Ä	リ	X	X	X	X	X	X	X	X	X	X	X
89	Y	Y	Y	Y	Y	Y	Y	ì	ル	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
90	Z	Z	Z	Z	Z	Z	Z	Ö	レ	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z	Z
91	[	[	[	Ø	[	[	[	Ü	ロ	[	Ä	Ä	Æ	Ä	'	[	'	¡	Ã	Æ
92	\	√	ç	Æ	¡	\	¥	É	ワ	\	Ö	Ö	Ø	Ö	ç	\	ç	Ñ	Ç	Ø
93	]	]	]	ø	]	]	]	ï	ン	]	Å	Å	Å	Ü	§	]	é	¿	Õ	Å
94	^	↑	^	æ	^	^	^	ß	°	^	^	Ü	^	^	^	^	^	^	^	^
95	_	_	_	_	_	_	_	Ô	°	_	_	_	_	_	_	_	_	_	_	_
96	`	`	`	`	`	`	`	Á		`	`	é	`	`	`	`	ù	`	`	`

ManualsPlus.com

Printing Characters, Variable-Space Fonts (Continued)

DECIMAL VALUE	SET 10	11	12	13	14	15	16	17	18	19	40	41	42	43	44	45	46	47	48	49
97	a	a	a	a	a	∩	a	Ã		a	a	a	a	a	a	a	a	a	a	a
98	b	b	b	b	b	⊃	b	ã		b	b	b	b	b	b	b	b	b	b	b
99	c	c	c	c	c	⊂	c	Đ		c	c	c	c	c	c	c	c	c	c	c
100	d	d	d	d	d	∪	d	đ		d	d	d	d	d	d	d	d	d	d	d
101	e	e	e	e	e	‾	e	Í		e	e	e	e	e	e	e	e	e	e	e
102	f	f	f	f	f	≡	f	Ì		f	f	f	f	f	f	f	f	f	f	f
103	g	g	g	g	g	≅	g	Ó		g	g	g	g	g	g	g	g	g	g	g
104	h	h	h	h	h	≈	h	Ò		h	h	h	h	h	h	h	h	h	h	h
105	i	i	i	i	i	~	i	Õ		i	i	i	i	i	i	i	i	i	i	i
106	j	j	j	j	j	≤	j	õ		j	j	j	j	j	j	j	j	j	j	j
107	k	k	k	k	k	≥	k	Š		k	k	k	k	k	k	k	k	k	k	k
108	l	l	l	l	l	≠	l	š		l	l	l	l	l	l	l	l	l	l	l
109	m	m	m	m	m	Δ	m	Ú		m	m	m	m	m	m	m	m	m	m	m
110	n	n	n	n	n	∏	n	Ÿ		n	n	n	n	n	n	n	n	n	n	n
111	o	o	o	o	o	Σ	o	ÿ		o	o	o	o	o	o	o	o	o	o	o
112	p	p	p	p	p	±	p	Þ		p	p	p	p	p	p	p	p	p	p	p
113	q	q	q	q	q	∓	q	þ		q	q	q	q	q	q	q	q	q	q	q
114	r	r	r	r	r	→	r			r	r	r	r	r	r	r	r	r	r	r
115	s	s	s	s	s	↑	s			s	s	s	s	s	s	s	s	s	s	s
116	t	t	t	t	t	←	t			t	t	t	t	t	t	t	t	t	t	t
117	u	u	u	u	u	↓	u			u	u	u	u	u	u	u	u	u	u	u
118	v	v	v	v	v	∫	v	—		v	v	v	v	v	v	v	v	v	v	v
119	w	w	w	w	w	÷	w	¼		w	w	w	w	w	w	w	w	w	w	w
120	x	x	x	x	x	*	x	½		x	x	x	x	x	x	x	x	x	x	x
121	y	y	y	y	y	∇	y	ª		y	y	y	y	y	y	y	y	y	y	y
122	z	z	z	z	z	·	z	º		z	z	z	z	z	z	z	z	z	z	z
123	{	↱					{	{	«	{	ä	ä	æ	ä	é	{	à	'	ã	æ
124	|	⊢					|	|	□	|	ö	ö	ø	ö	ù	|	ò	ñ	ç	ø
125	}	→					}	}	»	}	å	å	å	ü	è	}	è	ç	õ	å
126	~	~					~	~	±	—	—	ü	—	ß	¨	—	ì	~	·	~

ManualsPlus.com

Default Conditions Established by the DF Instruction

The following table shows default conditions established by the DF instruction. Additional conditions are listed after the table.

Function	Equivalent Instruction	Default Condition
Pen control	AP;	Automatic as follows: • Lift or store a motionless pen after 15 seconds for transparency fiber-tip pens or drafting pens, or after 65 seconds for paper fiber-tip pens and roller-ball pens • Select pen only when required to draw
Label buffer	BL ETX	Cleared
Alternate set	CA 0;	Character set 0
Character selection mode	CM;	HP 7-bit mode
Standard set	CS 0;	Character set 0
Chord tolerance	CT;	Set to angle mode for AA, AR, CI, and WG instructions
Character chord	CC;	Set variable-space font chord angle to 5 degrees
Digitize clear	DC;	Clear DP instruction and return to current display
Downloadable character buffer	DL;	Cleared
Relative direction	DR 1,0;	Horizontal characters
Label terminator	DT;	ETX (decimal code 3)
Extra space	ES 0,0;	No extra space between characters
Fill type, spacing, and angle	FT;	• Type 1, solid bidirectional fill • 1% of the diagonal distance between P1 and P2 • 0 degrees
Mask value	IM 223,0,0;	Recognizes all defined errors

(Table continued)

ManualsPlus.com

Function	Equivalent Instruction	Default Condition
Input window	IW;	Set to hard-clip limits
Label origin	LO 1;	Standard labeling starting at current position
Line type and pattern length	LT;	• Type 1, solid line • 4% of the diagonal distance between P1 and P2
Plotting mode	PA;	Absolute
Polygon mode	PM0; PM2;	Polygon buffer cleared
Pen thickness	PT;	0.3 mm
Scaling	SC;	User-unit scaling off
Character slant	SL 0;	0 degrees
Symbol mode	SM;	Off
Relative size	SR;	• Character width = 0.75% of $\lvert P2_X - P1_X \rvert$ • Character height = 1.5% of $\lvert P2_Y - P1_Y \rvert$
Select set	SS;	Select standard character set
Tick length	TL;	tp = tn = 0.5% of $\lvert P2_X - P1_X \rvert$ for Y-tick and 0.5% of $\lvert P2_Y - P1_Y \rvert$ for X-tick
User-defined fill type	UF;	Solid bidirectional fill

The following conditions are also established:

- The carriage-return point for labeling instructions is updated to the current pen position.
- PD and PU instructions with parameters are defaulted to be forms of the PA instruction.
- Although character size is defaulted as if "SI;" were executed, subsequent changes to the scaling points P1 and P2 will cause the character size to vary as if "SR;" were executed.

The following plotter functions are **not** affected by a DF instruction:

- Locations of P1 and P2

ManualsPlus.com

- Current pen and its position
- Pen speed, force, and acceleration
- 90-degree rotation or axis alignment
- Curved line generator (CV instruction)
- Setting of these front-panel conditions: **AUTO FEED** key, standard/enhanced, HP-IB address, eavesdrop/standalone, handshake, modem/direct, full/half duplex, parity, 7-bit/8-bit, and baud rate
- Function key definitions established by the KY and WD instructions

Default Conditions Established by the IN Instruction

The initialize instruction sets the plotter to the same conditions as the default instruction, DF, and sets these additional conditions:

- Raises the pen graphically and physically.
- Cancels 90-degree rotation.
- Sets default pen speed, force, and acceleration values in accordance with the carousel installed in the plotter.
- Sets bit position 3 of the status word to 1 (to indicate the plotter has been initialized).
- Clears any HP-GL (graphics) error condition.
- Clears lost mode (refer to Relationship of Plotting Instructions and Graphics Limits in Chapter 4).
- Clears the display and removes any function key definitions established by the WD and KY instructions.
- Sets the group count to 0 (GC instruction).
- Turns off the curved line generator (CV instruction).
- Returns P1, P2, and the axis-align point to the X,Y coordinate values that were set when the paper limits were established. Remember that any existing axis alignment is maintained.

NOTE: If an axis alignment has been set, only P1 will return to its default physical position on the paper. The hard-clip limits are still compressed, and P2 and the axis-align point will be rotated from their default physical positions by an amount corresponding to the axis alignment. Axis alignment is described in the Operation and Interconnection Manual. ■

ManualsPlus.com

An IN instruction is equivalent to switching the plotter off and then on again, except that axis alignment is not changed. Remember that the following front-panel functions are stored in the plotter's continuous memory; neither an IN instruction, nor switching power off and on, affects the current setting of these functions: **AUTO FEED** key, standard/enhanced, HP-IB address, eavesdrop/standalone, handshake, modem/direct, full/half duplex, parity, 7-bit/8-bit, and baud rate.

IN Instruction with the −1 Parameter

An IN −1 instruction resets the plotter to all of the above conditions with one exception. It does **not** reset the pen control (AP) conditions.

Default P1 and P2 Coordinates

These P1 and P2 coordinates are established when the plotter is turned on, or when an IN; or IP; instruction is executed. In each case the coordinates depend on the size of paper currently loaded.

Paper Size	Default (in Plotter Units)	
	$P1_X$,$P1_Y$	$P2_X$,$P2_Y$
A3	380,430	15580,10430
A4	430,200	10430,7400
A	80,320	10080,7520
B	620,80	15820,10080

NOTE: If an IP instruction is executed without parameters after the coordinate system has been rotated 90 degrees by the front-panel **ROTATE-90** function key or the RO90; instruction, the default locations of P1 and P2 are changed to reflect the rotation. The X- and Y-coordinates for each scaling point are reversed. For example, IP; after a 90-degree rotation on A-size paper sets P1 to be 320,80 and P2 to be 7520,10080. ■

The No Operation (NOP) Instructions

To maintain software compatibility with other HP plotters, the HP 7550 recognizes seven HP-GL instructions as no operation (NOP) instructions. If these instructions are included in a program, they are recognized by the HP 7550 and ignored without generating an error.

ManualsPlus.com

EC Enable Cutter
IC Input Character
OB Output Character Box Dimensions*
VA Adaptive Velocity
VN Normal Velocity

Scaling without Using the SC Instruction

The HP 7550 plotter movements are in terms of plotter units where a plotter unit = 0.025 mm. While the plotter can be scaled into user units using the SC instruction, it might be more convenient for you to write programs where numbers to be plotted are in some units other than plotter units or conventional user units. If so, you can use the following equations in your program to convert these "other units" into plotter units for the plotter:

$$X_{scaled} = \left[\frac{P2_X - P1_X}{U2_X - U1_X}\right] A_X + P1_X - U1_X \left[\frac{P2_X - P1_X}{U2_X - U1_X}\right]$$

$$Y_{scaled} = \left[\frac{P2_Y - P1_Y}{U2_Y - U1_Y}\right] A_Y + P1_Y - U1_Y \left[\frac{P2_Y - P1_Y}{U2_Y - U1_Y}\right]$$

where: A_X is the X-coordinate of the desired point in "user units,"

A_Y is the Y-coordinate of the desired point in "user units,"

$P1_X$ is the X-coordinate of P1 in plotter units,

$P1_Y$ is the Y-coordinate of P1 in plotter units,

$P2_X$ is the X-coordinate of P2 in plotter units,

$P2_Y$ is the Y-coordinate of P2 in plotter units,

$U1_X$ is the X-coordinate of P1 in "user units,"

$U1_Y$ is the Y-coordinate of P1 in "user units,"

$U2_X$ is the X-coordinate of P2 in "user units," and

$U2_Y$ is the Y-coordinate of P2 in "user units."

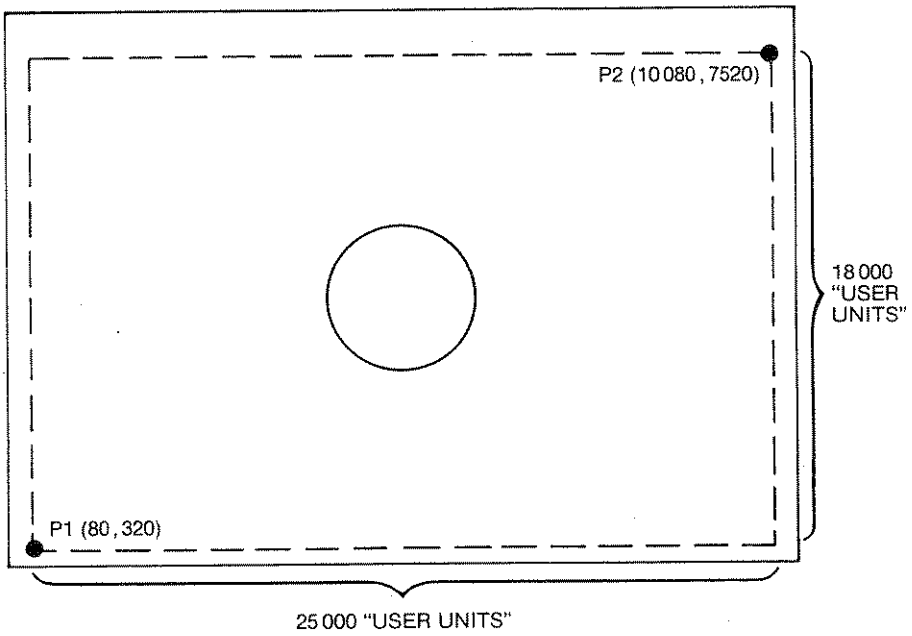
To demonstrate the use of the scaling equations, let's go through an example.

*The OB instruction is not functional, but will return four zeroes so that software will not hang waiting for a reply.

ManualsPlus.com

Problem

Scale the platen area (P1 = 80, 320 and P2 = 10 080, 7520 plotter units) into new "user units" where P1 = 0, 0 and P2 = 25 000, 18 000. At the center point (X = 12 500 and Y = 9000 "user units"), draw a circle with radius 2500 as shown below.



Solution

A. Recall that the equations of a circle are:

$$X = R \cos t$$
$$Y = R \sin t$$
where $0 \leqslant t \leqslant 2\pi$

B. Since we are to plot relative to a point that is not at the origin, an offset X_O, Y_O must be added to the circle equations. The offset in "user units" is:

$$X_O = 12\,500$$
$$Y_O = 9000$$

C. The desired circle equations are then:

$$A_X = 2500 \cos t + 12\,500$$
$$A_Y = 2500 \sin t + 9000$$

ManualsPlus.com

D. Determine the user scale:

$X = 0$ to $25\,000$
$Y = 0$ to $18\,000$

therefore

$U1_X = 0$
$U1_Y = 0$
$U2_X = 25\,000$
$U2_Y = 18\,000$

E. Determine the values for P1 and P2 which are set using the IP instruction:

$P1 = 80, 320$
$P2 = 10\,080, 7520$

therefore

$P1_X = 80$
$P1_Y = 320$
$P2_X = 10\,080$
$P2_Y = 7520$

F. Solving for X and Y, determine the equivalent plotter-unit values for the circle equations:

$$X = \left[\frac{P2_X - P1_X}{U2_X - U1_X}\right] A_X + P1_X - U1_X \left[\frac{P2_X - P1_X}{U2_X - U1_X}\right]$$

$$= \left[\frac{10\,080 - 80}{25\,000 - 0}\right] (2500 \cos t + 12\,500) + 80 - 0 \left[\frac{10\,080 - 80}{25\,000 - 0}\right]$$

$$= 0.4\,(2500 \cos t + 12\,500) + 80 - 0$$

$$= 1000 \cos t + 5080$$

$$Y = \left[\frac{P2_Y - P1_Y}{U2_Y - U1_Y}\right] A_Y + P1_Y - U1_Y \left[\frac{P2_Y - P1_Y}{U2_Y - U1_Y}\right]$$

$$+ \left[\frac{7520 - 320}{18\,000 - 0}\right] (2500 \sin t + 9000) + 320 - 0 \left[\frac{7520 - 320}{18\,000 - 0}\right]$$

$$= 0.4\,(2500 \sin t + 9000) + 320 - 0$$

$$= 1000 \sin t + 3920$$

ManualsPlus.com

G. The following program will plot the required circle using the equivalent plotter-unit values, and the default P1 and P2.

```
10  'Insert configuration statement here
20  PRINT #1, "IP80,320,10080,7520;SP1;"
25  PI=3.141593
30  FOR T=0 TO 2*PI+PI/20 STEP PI/20
40    X=1000*COS(T)+5080
50    Y=1000*SIN(T)+3920
60    PRINT #1, "PA";X;",";Y;";PD;"
70  NEXT T
80  PRINT #1, "SP0;"
90  END
```

The Scale Instruction, SC†

USES: The SC instruction establishes a user-unit coordinate system by mapping values onto the scaling points P1 and P2. Thus, you can plot in user units convenient to your application. For a discussion on the concept of scaling, refer to Scaling in Chapter 2.

SYNTAX: *SC* X_{min}, X_{max}, Y_{min}, Y_{max} *term*
or
SC *term*

Parameter	Format	Range	Default
X- and Y-ranges	integer	-2^{23} to $2^{23}-1$ user units	none

EXPLANATION: When you use parameters, you must specify all four of them. Note that the first two parameters are the X-axis range, and the last two parameters are the Y-axis range. As a result, the first and third parameters (X_{min} and Y_{min}) are the coordinate pair that is mapped onto P1; the second and fourth parameters (X_{max} and Y_{max}) are the coordinate pair that is mapped onto P2. (This is different from the IP instruction, where the parameters are expressed as X,Y coordinate pairs rather than as ranges.)

As their names suggest, you will normally want to specify X_{min} smaller than X_{max}, and Y_{min} smaller than Y_{max}. However, it is permissible to specify X_{min} larger than X_{max} and Y_{min} larger than Y_{max}. (If you do this, your plot could be drawn as a mirror image — reversed and/or upside down — depending also on the relative positions of P1 and P2.) X_{min} cannot be set equal to X_{max}, and Y_{min} cannot be set equal to Y_{max}.

†This instruction applies to plotters whose serial number prefix is 2520A or lower.

ManualsPlus.com

The parameters of the SC instruction are always mapped onto the current P1 and P2 locations. P1 and P2 retain these new values until scaling is turned off or another SC instruction redefines the user-unit values. Thus, the size of a user unit could change if any change is made in the relative position and distance between P1 and P2 *after* an SC instruction is executed.

While scaling is on (i.e., after an SC instruction has been executed), only those plotting instructions that can be issued in "current units" are interpreted as user units; all instructions that can only be issued in plotter units are still interpreted as plotter units. (The table in the SYNTAX section of each instruction tells you what kind of units each parameter requires.)

An SC instruction without parameters (SC;) turns off user-unit scaling; all subsequent plotting instructions will then be interpreted in plotter units.

The following table summarizes the possible error conditions or unexpected results that you might observe with the SC instruction.

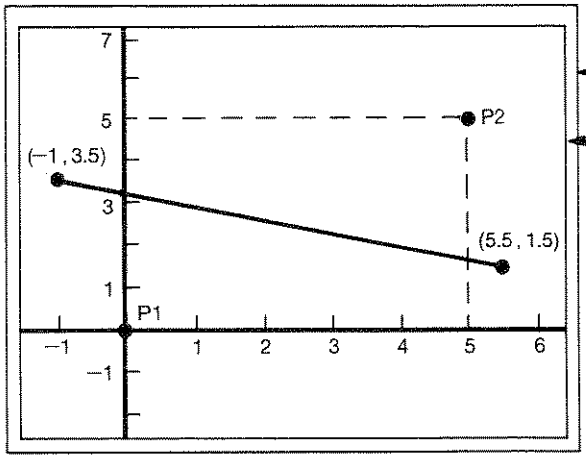
Condition	Error	Plotter Response
no parameters	none	turns scaling off
more than 4 parameters	2	executes first 4 parameters
less than 4 parameters	2	ignores instruction
$X_{min} = X_{max}$ or $Y_{min} = Y_{max}$ or parameter out-of-range	3	ignores instruction

Examples — Establishing Scaling

Remember that the SC parameters are mapped onto the current locations of P1 and P2. P1 and P2 do not represent a graphic limit; therefore, the new user-unit coordinate system extends across the entire range of the plotter-unit coordinate system. Thus, you can plot to a point beyond P1 or P2, as long as you are within the hard-clip limits. For example, you can plot from the point −1 , 3.5 to the point 5.5 , 1.5. This is illustrated on the next page.

ManualsPlus.com

```
"IP 2000,2000,8400,6000;"
"SC 0,5,0,5;"
```



A-SIZE PAPER EDGE

HARD-CLIP LIMITS

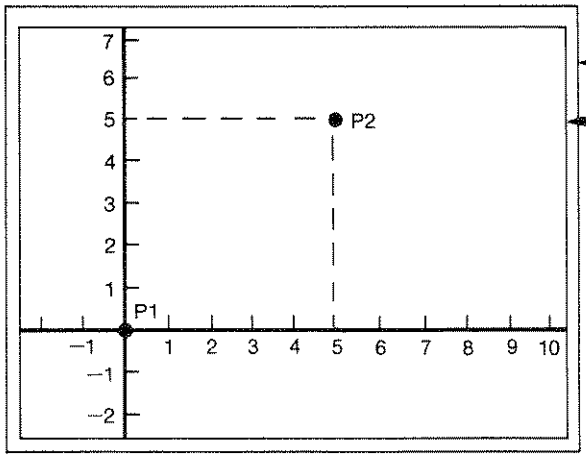
P1 = 0,0 (USER UNITS)
OR
2000,2000 (PLOTTER UNITS)

P2 = 5,5 (USER UNITS)
OR
8400,6000 (PLOTTER UNITS)

Another thing to consider when establishing user-unit scaling is whether your plots will include geometric or actual shapes. If so, you will need to establish isotropic scaling. Isotropic means the spacing of units along the X-axis is equal to the spacing along the Y-axis. When the scaling is isotropic, shapes such as circles will be plotted accurately without distortion. In the previous illustration, the scaling was anisotropic, or unequal along the X- and Y-axes. If you wanted to plot a circle using that scaling, it would turn out as an ellipse (egg-shaped) instead of a circle.

The easiest way to establish isotropic scaling is to set P1 and P2 so that they define a square area. Then assign a scale that places the same number of units along the X-axis as along the Y-axis. This is shown below.

```
"IP 2000,2000,6000,6000;"
"SC 0,5,0,5;"
```



PAPER EDGE

HARD-CLIP LIMITS

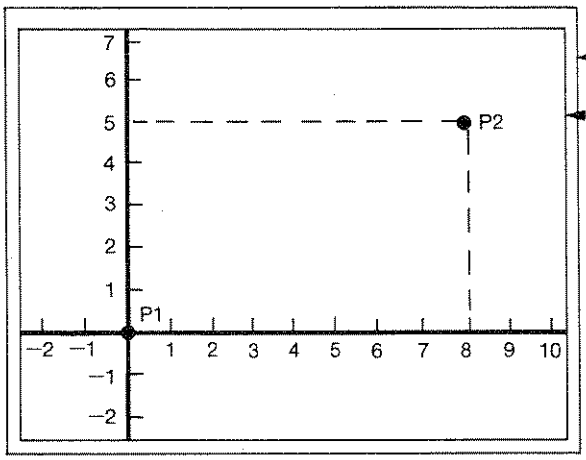
P1 = 0,0 (USER UNITS)
OR
2000,2000 (PLOTTER UNITS)

P2 = 5,5 (USER UNITS)
OR
6000,6000 (PLOTTER UNITS)

ManualsPlus.com

Another method for establishing isotropic scaling is to specify your scale so that each user unit equals the same number of plotter units in each direction. Using the P1/P2 locations that were set up in the first example, notice that the Y-axis distance from P1 to P2 is 4000 plotter units, or 5 user units. This means that each user unit is 800 plotter units (4000/5 = 800). To achieve this proportion along the X-axis, which is 6400 plotter units from P1 to P2, you must specify 8 user units (6400/800 = 8), as shown below. Compare this chart with the last one. Notice that both have the same number of units within the hard-clip limits even though P1 and P2 are in different locations.

```
"IP 2000,2000,8400,6000;"
"SC 0,8,0,5;"
```



ManualsPlus.com

Notes

ManualsPlus.com

Appendix B
Error Messages

If you execute an instruction that has errors in syntax or parameters, the plotter informs you by displaying an error message on the front panel. The error message remains on the front-panel display until you press a front-panel key or read the error number.

There are two types of errors: those related to HP-GL instructions, and those related to device-control instructions. To read an HP-GL error number in your program, use the OE instruction (presented in Chapter 13). To read a device-control error number in your program, use the ESC . E instruction (presented in Chapter 14).

The following tables list the front-panel error messages and their meanings.

HP-GL Errors

Error No.	Displayed Message	Usual Plotter Reaction	Possible Cause
1	Command not recognized	Ignores the instruction	A mnemonic is incorrect or missing; an alphabetic character was specified in a parameter when a numeric character was expected.*

*Check for typographical errors in the mnemonic and the parameters (a common mistake is to type O for 0, or vice versa). Also, check to be sure instructions are not being terminated prematurely. For example, if you are using an HP-IB configuration, check to be sure the computer is not inserting line feeds (LF) in the middle of a string of instructions. If an instruction is terminated after the mnemonic and before the parameters, the subsequent receipt of the parameters without a mnemonic will cause error 1.

(Table continued)

ManualsPlus.com

HP-GL Errors (Continued)

Error No.	Displayed Message	Usual Plotter Reaction	Possible Cause
2	Wrong number of parameters	If too few parameters, ignores the instruction. If too many parameters, executes the instruction with the correct number of parameters and ignores the rest.	Too few or too many parameters; an incomplete X,Y coordinate pair.
3	Bad parameter	Ignores the instruction	A parameter is out-of-range.**
4	(unused)	(unused)	(unused)
5	Unknown character set	Ignores the instruction	A set other than −1, 0–9, 10–19, 30–39, or 40–49 has been designated or invoked.
6	Position overflow	Ignores the instruction	A single label is so long that it exceeds the plotter's numeric range.
7	Buffer overflow	Excecutes the data that fits in the affected buffer and ignores the data that overflows	One of the graphics memory buffers does not have enough space allocated.***

**Check each parameter range for the instructions that are suspected of causing the error. Parameter ranges are listed at the beginning of each instruction's discussion throughout this manual, as well as in Appendix C.

***Check buffer allocations. Refer to The Allocate Configurable Memory Instruction, ESC.T, in Chapter 14 for information on the plotter's buffers.

ManualsPlus.com

Device-Control Errors

Error No.	Displayed Message	Meaning
0	(none)	No I/O error has occurred.
10	Invalid I/O output request	(*RS-232-C only*). New output has been generated before previous output was finished being transmitted. The previous output will continue normally and the new output will be ignored (thus causing the error).
11	Invalid byte folowing **ESC.**	Invalid character received after first two characters (**ESC.**) in a device-control instruction.
12	Invalid byte in I/O control	Invalid character received while parsing a device-control instruction. The parameter containing the invalid character and all following parameters are defaulted.
13	Out-of-range I/O parameter	One or more parameters are out-of-range.
14	Too many I/O parameters	Too many parameters received. Additional parameters beyond the proper number are ignored; parsing of the instruction ends when a colon (normal termination) or the next **ESC** character (abnormal termination) is received. **NOTE:** The receipt of a character other than another parameter, a semicolon, or a colon will result in error 12 overwriting error 14. ■
15	Error in I/O transmission	(*RS-232-C only*). A framing error, parity error, or overrun error has been detected. The defective character is replaced by **DEL** (decimal code 127).

(Table continued)

ManualsPlus.com

Device-Control Errors (Continued)

Error No.	Displayed Message	Meaning
16	I/O buffer overflow	(*RS-232-C only*). The physical I/O buffer has overflowed. As a result, one or more characters have been lost; therefore, an HP-GL error will probably occur. The last valid character is replaced by **DEL** (decimal code 127).
17	Transmit underrun	(*RS-232-C only*). Transmit underrun can be caused by a baud rate mismatch between devices, or by excessive I/O activity in receive monitor mode.
18	I/O error indeterminate	I/O error of indeterminate cause.

ManualsPlus.com

Appendix C
Instruction Summary

HP-GL Instructions

This section lists the formal syntax and parameter ranges for HP-GL instructions, in alphabetical order of the instruction's mnemonic. For more information about each instruction, refer to the indicated page number.

The semicolon is included as the terminator. (However, a semicolon or the next mnemonic are each valid terminators. In an HP-IB configuration, a line-feed character is also a valid terminator.) [TERM] means the terminator sent by the plotter at the end of an output response. It is [**CR LF**] in an HP-IB configuration, and [**CR**] or as set by the ESC . M instruction in an RS-232-C configuration.

AA The Arc Absolute Instruction

Page 6-9

AA X, Y, arc angle (, chord tolerance) ;

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none
arc angle	decimal	-2^{23} to $2^{23}-1$ degrees	none
chord tolerance	decimal	-2^{23} to $2^{23}-1$ current mode	5 degrees

AF or AH The Advance Page Instruction

Page 10-12

AF ;
or
AH ;

ManualsPlus.com

AP The Automatic Pen Operations Instruction Page 10-2

AP n;
or
AP ;

Parameter	Format	Range	Default
n	integer	0–143	7

AR The Arc Relative Instruction Page 6-9

AR X, Y, arc angle (, chord tolerance);

Parameter	Format	Range	Default
X- and Y-increments	decimal	-2^{23} to $2^{23}-1$ current units	none
arc angle	decimal	-2^{23} to $2^{23}-1$ degrees	none
chord tolerance	decimal	-2^{23} to $2^{23}-1$ current mode	5 degrees

AS The Acceleration Select Instruction Page 10-7

AS pen acceleration (, pen number);
or
AS ;

Parameter	Format	Range	Default
pen acceleration	integer	1–6	6
pen number	integer	1–8	all pens

BF The Buffer Plot Instruction Page 10-13

BF ;

ManualsPlus.com

BL The Buffer Label Instruction

Page 7-38

BL c ... c *term*
or
BL *term*
(where *term* is the label terminator defined by the DT instruction)

Parameter	Format	Range	Default
c ... c (up to 150 characters are buffered)	label	any character	none

CA The Designate Alternate Character Set Instruction

Page 11-4

CA set;
or
CA ;

Parameter	Format	Range	Default
set	integer	−1, 0–19, 30–49	0

CC The Character Chord Angle Instruction

Page 11-26

CC chord angle;
or
CC ;

Parameter	Format	Range	Default
chord angle	decimal	$\lvert -2^{23}$ to $2^{23}-1 \rvert$ degrees	5 degrees

CI The Circle Instruction

Page 6-5

CI radius (, chord tolerance);

Parameter	Format	Range	Default
radius	decimal	-2^{23} to $2^{23}-1$ current units	none
chord tolerance	decimal	-2^{23} to $2^{23}-1$ current mode	5 degrees

ManualsPlus.com

CM The Character Selection Mode Instruction Page 11-20

CM switch mode (, fallback mode) ;
or
CM ;

Parameter	Format	Range	Default
switch mode	integer	0–3	0
fallback mode	integer	0–1	0

CP The Character Plot Instruction Page 7-33

CP spaces, lines ;
or
CP ;

Parameter	Format	Range	Default
spaces	decimal	-2^{23} to $2^{23}-1$	none
lines	decimal	-2^{23} to $2^{23}-1$	none

CS The Designate Standard Character Set Instruction Page 11-4

CS set ;
or
CS ;

Parameter	Format	Range	Default
set	integer	−1, 0–19, 30–49	0

CT The Chord Tolerance Instruction Page 6-3

CT n ;
or
CT ;

Parameter	Format	Range	Default
n	integer	0 or 1	0

ManualsPlus.com

CV The Curved Line Generator Instruction

Page 10-10

CV n (, input delay) ;
or
CV ;

Parameter	Format	Range	Default
n	integer	0 or 1	0
input delay	integer	-2^{23} to $2^{23}-1$ milliseconds	100 ms

DC The Digitize Clear Instruction

Page 12-4

DC ;

DF The Default Instruction

Page 3-8

DF ;

See table in Chapter 3 or Appendix A.

DI The Absolute Direction Instruction

Page 7-21

DI run, rise ;
or
DI ;

Parameter	Format	Range	Default
run (cos θ)	decimal	-2^{23} to $2^{23}-1$	1
rise (sin θ)	decimal	-2^{23} to $2^{23}-1$	0

DL The Define Downloadable Character Instruction

Page 11-36

DL character number (, pen control), X,Y (,...) (, pen control) (,...) ;
or
DL character number ;
or
DL ;

(DL instruction continued)

ManualsPlus.com

Parameter	Format	Range	Default
character number	integer	33–126	none
pen control	integer	−128	none
X,Y coordinates	integer	−127–127 primitive grid units	none

DP The Digitize Point Instruction

Page 12-2

DP ;

DR The Relative Direction Instruction

Page 7-25

DR run, rise;
or
DR ;

Parameter	Format	Range	Default
run (cos θ)	decimal	-2^{23} to $2^{23}-1$	1% of $\lvert P2_X - P1_X \rvert$
rise (sin θ)	decimal	-2^{23} to $2^{23}-1$	0% of $\lvert P2_Y - P1_Y \rvert$

DS The Designate Character Set into Slot Instruction

Page 11-22

DS slot, set;
or
DS ;

Parameter	Format	Range	Default
slot	integer	0–1 (HP modes) 0–3 (ISO modes)	0
set	integer	−1, 0–19, 30–49	0

DT The Define Label Terminator Instruction

Page 7-8

DT label terminator;
or
DT ;

(DT instruction continued)

ManualsPlus.com

Parameter	Format	Range	Default
label terminator	label	any character except **NULL**, **LF**, **ESC** and ; (decimal codes 0, 10, 27, and 59, respectively)	**ETX** (decimal code 3)

EA The Edge Rectangle Absolute Instruction

Page 6-27

EA X-coordinate, Y-coordinate;

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none

EP The Edge Polygon Instruction

Page 6-43

EP ;

ER The Edge Rectangle Relative Instruction

Page 6-33

ER X-increment, Y-increment;

Parameter	Format	Range	Default
X- and Y-increments	decimal	-2^{23} to $2^{23}-1$ current units	none

ES The Extra Space Instruction

Page 7-36

ES spaces (, lines);
or
ES ;

Parameter	Format	Range	Default
spaces	decimal	-2^{23} to $2^{23}-1$	0
lines	decimal	-2^{23} to $2^{23}-1$	0

ManualsPlus.com

EW The Edge Wedge Instruction

Page 6-21

EW radius, start angle, sweep angle (, chord tolerance) ;

Parameter	Format	Range	Default
radius	decimal	-2^{23} to $2^{23}-1$ current units	none
start angle	decimal	-2^{23} to $2^{23}-1$ degrees, modulo 360	none
sweep angle	decimal	-2^{23} to $2^{23}-1$ degrees, truncated at ±360	none
chord tolerance	decimal	-2^{23} to $2^{23}-1$ current mode	5 degrees

FP The Fill Polygon Instruction

Page 6-44

FP ;

FS The Force Select Instruction

Page 10-6

FS pen force (, pen number) ;
or
FS ;

Parameter	Format	Range	Default
pen force	integer	1–8	depends on carousel type
pen number	integer	1–8	all pens

FT The Fill Type Instruction

Page 6-13

FT type (, spacing (, angle)) ;
or
FT ;

Parameter	Format	Range	Default
fill type	integer	1–6	1
spacing	decimal	0 to $2^{23}-1$ current units	depends on fill type
angle	decimal	-2^{23} to $2^{23}-1$ degrees, modulo 360	0 degrees

ManualsPlus.com

GC The Group Count Instruction

Page 10-25

GC count number;
or
GC ;

Parameter	Format	Range	Default
count number	integer	-2^{23} to $2^{23}-1$	0

GM The Graphics Memory Instruction

Page 3-18

GM (polygon buffer) (, downloadable character buffer) (, replot buffer) (, vector buffer);
or
GM ;

Parameter	Format	Range	Default
polygon buffer	integer	4–12 752 bytes	1778
downloadable character buffer	integer	0–12 754 bytes	0
replot buffer	integer	0–12 750 bytes	9954
vector buffer	integer	44–12 794 bytes	44

GP The Group Pen Instruction†

Page 10-4

GP (group number (,pen number (,number of pens (,length))));

Parameter	Format	Range	Default
group number	integer	1–8	all groups
pen number	integer	1–8	specified group number
number of pens	integer	1–8*	1
length	integer	1–50 000	100

*Pen number + number of pens must be < 9.

IM The Input Mask Instruction

Page 13-4

IM E-mask value (, S-mask value (, P-mask value));
or
IM ;

(IM instruction continued)

†This instruction applies to plotters whose serial number prefix is 2629A or larger.

ManualsPlus.com

Parameter	Format	Range	Default
E-mask value	integer	0–255	223
S-mask value	integer	0–255	0
P-mask value	integer	0–255	0

IN The Initialize Instruction

Page 3-11

IN ;

IP The Input P1 and P2 Instruction

Page 3-12

IP $P1_X$, $P1_Y$(, $P2_X$, $P2_Y$);
or
IP ;

Parameter	Format	Range	Default
X- and Y-coordinates	integer	-2^{23} to $2^{23}-1$ plotter units	depends on paper size

IV The Invoke Character Slot Instruction

Page 11-24

IV slot,(left);
or
IV ;

Parameter	Format	Range	Default
slot	integer	0–1 (HP modes) 0–3 (ISO modes)	0
left	integer	0–1	

IW The Input Window Instruction

Page 9-5

IW X_1, Y_1, X_2, Y_2;
or
IW ;

Parameter	Format	Range	Default
X- and Y-coordinates	integer	-2^{23} to $2^{23}-1$ current units if **ENHANCED** function key is on; plotter units if **STANDARD** function key is on	current hard-clip limits (depends on paper size)

ManualsPlus.com

KY The Define Key Instruction

Page 10-20

KY key (, function);
or
KY ;

Parameter	Format	Range	Default
key	integer	1–4	none
function	integer	0–12	none

LB The Label Instruction

Page 7-2

LB c ... c *term*
(where *term* is the label terminator defined by the DT instruction)

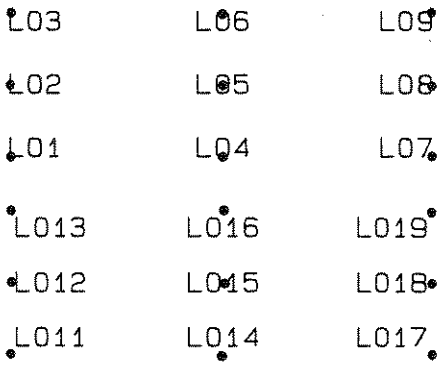
Parameter	Format	Range	Default
c ... c	label	any character	none

LO The Label Origin Instruction

Page 7-30

LO position number;
or
LO ;

Parameter	Format	Range	Default
position number	integer	1–9 or 11–19	1



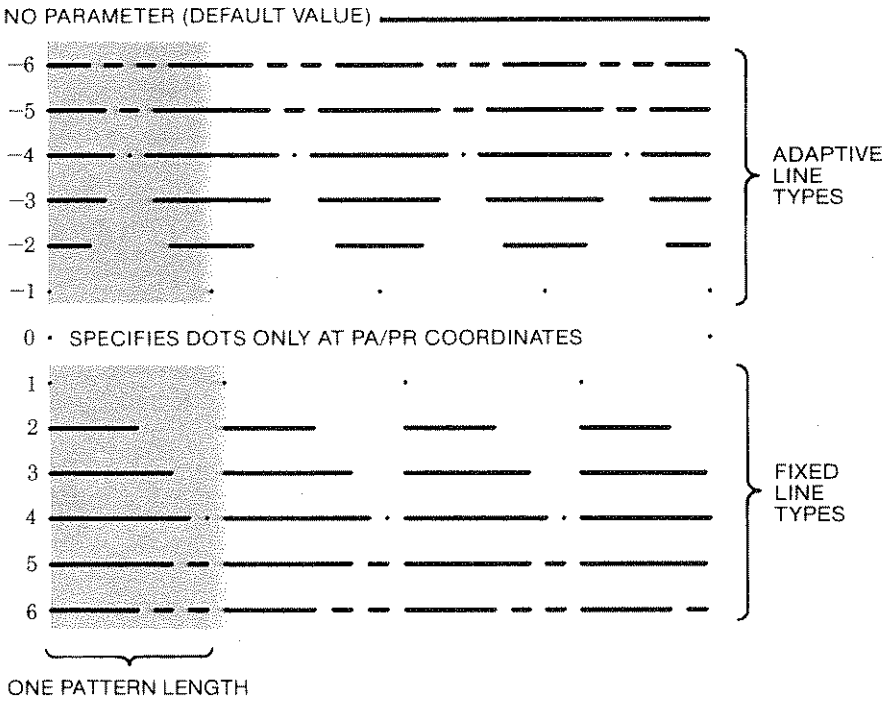
LT pattern number (, pattern length);
or
LT ;

(LT instruction continued)

ManualsPlus.com

Parameter	Format	Range	Default
pattern number	integer	−6 to +6	no parameter (solid line)
pattern length	decimal	0 to $2^{23} - 1$ percentage	4% of the diagonal distance between P1 and P2

The line types for each pattern number are shown below.



NR The Not-Ready Instruction

Page 10-14

NR ;

OA The Output Actual Position and Pen Status Instruction

Page 13-7

OA ;

Response: X,Y,P [TERM] — integers, in ASCII.

X,Y — in plotter units within current hard-clip limits.

P — 0, pen up or 1, pen down.

ManualsPlus.com

OC The Output Commanded Position and Pen Status Instruction

Page 13-8

OC ;

Response: X,Y, P [TERM] — two decimals and one integer, in ASCII.

X,Y — in current units, -2^{23} to $2^{23}-1$.

P — 0, pen up or 1, pen down.

OD The Output Digitized Point and Pen Status Instruction

Page 12-3

OD ;

Response: X,Y,P [TERM] — integers, in ASCII.

X,Y — in current units if **ENHANCED** function key is on;
in plotter units if **STANDARD** function key is on.

P — 0, pen up or 1, pen down.

OE The Output Error Instruction

Page 13-8

OE ;

Response: Error number [TERM] — a positive ASCII integer, 0 to 7 (refer to table in Appendix B).

OF The Output Factors Instruction

Page 13-9

OF ;

Response: 40 , 40 [TERM] — integers, in ASCII.

OG The Output Group Count Instruction

Page 10-26

OG ;

Response: Count number, escape status [TERM] — integers, in ASCII.

Count number — -2^{23} to $2^{23}-1$

Escape status — 0, **ESCAPE** function has not been activated;
−1, **ESCAPE** function has been activated.

OH The Output Hard-Clip Limits Instruction

Page 9-10

OH ;

Response: X_{LL}, Y_{LL}, X_{UR}, Y_{UR}, [TERM] — ASCII integers representing plotter units.

ManualsPlus.com

OI The Output Identification Instruction

Page 13-10

OI ;

Response: 7550A [TERM] — ASCII string, five characters.

OK The Output Key Instruction

Page 10-23

OK ;

Response: Function key pressed [TERM] — integer in ASCII, 0 to 4.

OL The Output Label Length Instruction

Page 7-40

OL ;

Response: Length, characters, line feeds [TERM] — in ASCII.

Length — longest line in the buffered label as a decimal number with three places to the left and four places to the right of the decimal. In terms of the space dimension of the CP cell.

Characters — integer, the number of printing characters and spaces in the longest line of the buffered label.

Line feeds — integer, net number of line feeds.

OO The Output Options Instruction

Page 13-10

OO ;

Response: C,1,0,0,1,1,0,1 [TERM] — integers in ASCII.

C — 0 to 3, representing status of paper check and paper feed bits.

OP The Output P1 and P2 Instruction

Page 9-11

OP ;

Response: $P1_X$, $P1_Y$, $P2_X$, $P2_Y$ [TERM] — ASCII integers representing plotter units.

OS The Output Status Instruction

Page 13-11

OS ;

Response: Status [TERM] — integer in ASCII, 0 to 255. Power-on status, 26.

ManualsPlus.com

OT The Output Carousel Type Instruction

Page 13-12

OT ;

Response: Type, map [TERM] — integers, in ASCII.

Type — −1 to 4.

Map — 0 to 255.

OW The Output Window Instruction

Page 9-9

OW ;

Response: $X_{LL}, Y_{LL}, X_{UR}, Y_{UR}$ [TERM] — integers, in ASCII.

X, Y — in current units if **ENHANCED** function key is on;
in plotter units if **STANDARD** function key is on.

PA The Plot Absolute Instruction

Page 4-4

PA X, Y (,...);
or
PA ;

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none

PB The Print Buffered Label Instruction

Page 7-39

PB ;

PD The Pen Down Instruction

Page 4-3

PD X, Y (,...);
or
PD ;

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none

ManualsPlus.com

PG The Page Feed Instruction

Page 10-12

PG n;
or
PG ;

Parameter	Format	Range	Default
n	integer	-2^{23} to $2^{23}-1$	none

PM The Polygon Mode Instruction

Page 6-37

PM n;

Parameter	Format	Range	Default
n	integer	0–2	0

PR The Plot Relative Instruction

Page 4-6

PR X,Y (,...);
or
PR ;

Parameter	Format	Range	Default
X- and Y-increments	decimal	-2^{23} to $2^{23}-1$ current units	none

PT The Pen Thickness Instruction

Page 6-20

PT pen thickness;
or
PT ;

Parameter	Format	Range	Default
pen thickness	decimal	0.1–5.0 millimetres	0.3

ManualsPlus.com

PU The Pen Up Instruction

Page 4-3

PU X,Y (,...);
or
PU ;

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none

RA The Fill Rectangle Absolute Instruction

Page 6-27

RA X-coordinate, Y-coordinate;

Parameter	Format	Range	Default
X- and Y-coordinates	decimal	-2^{23} to $2^{23}-1$ current units	none

RO The Rotate Coordinate System Instruction

Page 9-2

RO n;
or
RO ;

Parameter	Format	Range	Default
n	integer	0 or 90 degrees	0

RP The Replot Instruction

Page 10-16

RP n;

Parameter	Format	Range	Default
n	integer	1-99	1

RR The Fill Rectangle Relative Instruction

Page 6-33

RR X-increment, Y-increment;

(RR instruction continued)

ManualsPlus.com

Parameter	Format	Range	Default
X- and Y-increments	decimal	-2^{23} to $2^{23}-1$ current units	none

SA The Select Alternate Character Set Instruction

Page 11-9

SA ;

SC The Scale Instruction†

Page 3-14

SC X_{min}, X_{max}, Y_{min}, Y_{max} (, type (, left , bottom)) ;
or
SC X_{min}, X_{factor}, Y_{min}, Y_{factor}, 2 ;

Parameter	Format	Range	Default
X_{min}	decimal	-2^{23} to $2^{23}-1$	none
X_{max}	decimal	-2^{23} to $2^{23}-1$	none
Y_{min}	decimal	-2^{23} to $2^{23}-1$	none
Y_{max}	decimal	-2^{23} to $2^{23}-1$	none
type	integer	0 and 1	0
left	decimal	0 to 100%	50%
bottom	decimal	0 to 100%	50%
X_{factor}	decimal	0.1 to $2^{23}-1$	none
Y_{factor}	decimal	0.1 to $2^{23}-1$	none

SG The Select Pen Group Instruction††

Page 10-6

SG group number ;

Parameter	Format	Range	Default
group number	integer	0 to 8	0 (no pen)

†This instruction applies to plotters whose serial number prefix is 2629A or higher. If the serial number prefix is 2520A or lower, refer to Appendix A for the SC instruction for your plotter.

††This instruction applies to plotters whose serial number prefix is 2629A or larger.

ManualsPlus.com

SI The Absolute Character Size Instruction

Page 7-14

SI width, height;
or
SI ;

Parameter	Format	Range	Default
width	decimal	-2^{23} to $2^{23}-1$ centimetres*	0.285 cm (A3/B-size paper) 0.187 cm (A4/A-size paper)
height	decimal	-2^{23} to $2^{23}-1$ centimetres*	0.375 cm (A3/B-size paper) 0.269 cm (A4/A-size paper)

*excluding zero (0) and values approaching zero

SL The Character Slant Instruction

Page 7-18

SL tan θ;
or
SL ;

Parameter	Format	Range	Default
tangent θ	decimal	±0.05 to ±2 for default characters ±0.05 to ±3.5 for large characters	0 (no slant)

SM The Symbol Mode Instruction

Page 5-5

SM character;
or
SM ;

Parameter	Format	Range	Default
character	label	any printing character (decimal codes 33–126)	none

ManualsPlus.com

SP The Select Pen Instruction

Page 4-2

SP pen number;
or
SP ;

Parameter	Format	Range	Default
pen number	integer	0–8	0

SR The Relative Character Size Instruction

Page 7-16

SR width, height;
or
SR ;

Parameter	Format	Range	Default
width	decimal	-2^{23} to $2^{23}-1$ percentage*	0.75% of $\lvert P2_X - P1_X \rvert$
height	decimal	-2^{23} to $2^{23}-1$ percentage*	1.5% of $\lvert P2_Y - P1_Y \rvert$

*excluding zero (0) and values approaching zero

SS The Select Standard Character Set Instruction

Page 11-9

SS ;

TL The Tick Length Instruction

Page 5-2

TL tp (, tn);
or
TL ;

Parameter	Format	Range	Default
tp and tn	decimal	0 to $2^{23}-1$ percentage	0.5% of $\lvert P2_X - P1_X \rvert$ and of $\lvert P2_Y - P1_Y \rvert$

ManualsPlus.com

UC The User-Defined Character Instruction

Page 11-28

UC (pen control,) X-increment, Y-increment (, pen control) (,...);
or
UC ;

Parameter	Format	Range	Default
pen control	integer	**STANDARD:*** ⩾ +99 = pen down ⩽ −99 = pen up **ENHANCED:*** ⩾ +9999 = pen down ⩽ −9999 = pen up	pen up
X- and Y increments	integer	**STANDARD:*** −98 to 98 **ENHANCED:*** −9998 to 9998 (both in primitive grid units)	none

*The ranges depend on the setting of the front-panel **STANDARD/ENHANCED** function key, as shown. The pen control parameters cannot exceed the plotter's range of -2^{23} to $2^{23}-1$.

UF The User-Defined Fill Type Instruction

Page 6-17

UF gap_1 (, gap_2,...gap_{20});
or
UF ;

Parameter	Format	Range	Default
gap	integer	0 to $2^{23}-1$	none

VS The Velocity Select Instruction

Page 10-8

VS pen speed (, pen number);
or
VS ;

Parameter	Format	Range	Default
pen speed	integer	1–80	depends on carousel type
pen number	integer	1–8	all pens

ManualsPlus.com

WD The Write to Display Instruction

Page 10-18

WD c ... c *term*
or
WD *term*
(where *term* is the label terminator defined by the DT instruction)

Parameter	Format	Range	Default
c ... c (up to 32)	label	any character from decimal code 32 to 95	none

WG The Fill Wedge Instruction

Page 6-21

WG radius, start angle, sweep angle (, chord tolerance);

Parameter	Format	Range	Default
radius	decimal	-2^{23} to $2^{23}-1$ current units	none
start angle	decimal	-2^{23} to $2^{23}-1$ degrees, modulo 360	none
sweep angle	decimal	-2^{23} to $2^{23}-1$ degrees, truncated at ± 360	none
chord tolerance	decimal	-2^{23} to $2^{23}-1$ current mode	5 degrees

XT The X-Tick Instruction

Page 5-1

XT ;

YT The Y-Tick Instruction

Page 5-1

YT ;

Device-Control Instructions

This section lists the formal syntax for device-control instructions in alphabetical order of the escape sequence. All instructions apply to both the HP-IB and RS-232-C/CCITT V.24 configurations unless otherwise noted in the title. Refer to the indicated page number for details.

ManualsPlus.com

Set Plotter Configuration

Page 14-12

ESC . @ [(<DEC>);(<DEC>)]:

Parameters: <DEC> — Specifies logical I/O buffer size (2 to 12752 bytes).

Decimal equivalent value 0 to 127. Bits 0, 1, and 4 apply to RS-232-C only.

Bit 0. Logic state 0: Disable hardwire handshake (ignore DTR line, pin 20). Logic state 1: Enable hardwire handshake (utilize DTR line, pin 20).

Bit 1. Logic state 0: Computer holds off data from the plotter using the CTS and DSR lines (pins 5 and 6). Logic state 1: Computer does not hold off data from the plotter using the CTS and DSR lines (pins 5 and 6).

Bit 2. Logic state 0: Specify parse monitor mode. Logic state 1: Specify receive monitor mode.

Bit 3. Logic state 0: Disable monitor mode. Logic state 1: Enable the monitor mode specified by bit 2.

Bit 4. Logic state 0: Disable block I/O error checking. Logic state 1: Enable block I/O error checking.

Output Identification

Page 14-27

ESC . A

Response: <ASC>,<DEC> [TERM] — 7550A (ASCII string), firmware revision level (integer).

Output Buffer Space

Page 14-15

ESC . B

Response: <DEC> [TERM] — 2 to 12752 bytes, the number of unused bytes in the logical I/O buffer.

Output Extended Error

Page 14-16

ESC . E

Response: <DEC> [TERM] — 0 (no error) or 10 to 18 (refer to table in Appendix B).

ManualsPlus.com

Set Handshake Mode 1 (RS-232-C only) Page 16-35

ESC. H [(<DEC>);(<ASC>);(<ASC>(;...<ASC>))]:

Parameters: <DEC> — Data block size, 0 to 32 767.

<ASC> — Enquiry character, ASCII 0 to 126.

<ASC>...<ASC> — Acknowledgment string of one to 10 characters, ASCII 0 to 127.

Set Handshake Mode 2 (RS-232-C only) Page 16-37

ESC. I [(<DEC>);(<ASC>);(<ASC>(;...<ASC>))]:

Parameters: <DEC> — Data block size or Xoff threshold level.

<ASC> — Enquiry character or omitted for Xon-Xoff.

<ASC>...<ASC> — Xon trigger character(s) or acknowledgment string of one to 10 characters.

Independent of Set Output Mode Instruction, ESC. M.

Abort Device Control (RS-232-C only) Page 16-39

ESC. J

Abort Graphics Page 14-25

ESC. K

Output Buffer Size When Empty Page 14-11

ESC. L

Response: <DEC> [TERM] — 2 to 12 752 bytes; not output until the logical I/O buffer is empty.

Set Output Mode (RS-232-C only) Page 16-28

ESC. M [(<DEC>);(<ASC>);(<ASC>);
(<ASC>(;<ASC>));(<ASC>)]:

Parameters: <DEC> — Turnaround delay, 0 to 9999 milliseconds.

<ASC> — Output trigger character, ASCII 0 to 126.

(ESC. M instruction continued)

ManualsPlus.com

<ASC> — Echo terminate character, ASCII 0 to 126.

<ASC>...<ASC> — one or two output terminator characters ASCII 0 to 127.

<ASC> — Output initiator character, ASCII 0 to 127.

Set Extended Output and Handshake Mode (RS-232-C only)

Page 16-31

ESC.N [(<DEC>);(<ASC>(;...<ASC>))]:

Parameters: <DEC> — Intercharacter delay, 0 to 9999 milliseconds.

<ASC>...<ASC> — Xoff trigger character(s) or immediate response string; one to 10 characters, ASCII 0 to 127, 0 terminates string.

Output Extended Status

Page 14-19

ESC.O

Response: <DEC> [TERM] — Status, decimal equivalent value 0–1775.

Set Handshake Mode (RS-232-C only)

Page 16-27

ESC.P (<DEC>):

Parameter: <DEC> — Selects standard handshake:

0 none
1 Xon-Xoff
2 ENQ/ACK
3 hardwire

Set Monitor Mode

Page 14-22

ESC.Q (<DEC>):

Parameter: <DEC> — 0 disables monitor mode
1 enables parse monitor mode
2 enables receive monitor mode

Reset

Page 14-26

ESC.R

NOTE: Should be followed by ESC.L. ■

ManualsPlus.com

Output Configurable Memory Size

Page 14-11

ESC. S (<DEC>):

Parameter: <DEC> — 0 requests total size of configurable graphics memory
1 requests current physical I/O buffer size
2 requests current polygon buffer size
3 requests current downloadable character buffer size
4 requests current replot buffer size
5 requests current vector buffer size

Response: <DEC> [TERM] — 0 to 12800 bytes.

Allocate Configurable Memory

Page 14-5

ESC. T [(<DEC>);(<DEC>);(<DEC>);(<DEC>);(<DEC>)]:

Parameters: <DEC> — physical I/O buffer size: 2 to 12752
<DEC> — polygon buffer size: 4 to 12754
<DEC> — downloadable character buffer size: 0 to 12750
<DEC> — replot buffer size: 0 to 12750
<DEC> — vector buffer size: 44 to 12794

NOTE: Should be followed by ESC.L. ■

End Flush Mode

Page 14-27

ESC. U

Plotter-On (RS-232-C only)

Page 16-39

ESC. Y or **ESC.**(

Plotter-Off (RS-232-C only)

Page 16-40

ESC. Z or **ESC.**)

ManualsPlus.com

Subject Index

a

AA Instruction ..... 6-9 thru 6-13, C-1
AF Instruction ..... 10-12 thru 10-14, C-1
AGL 1-6
AH Instruction ..... 10-12 thru 10-14, C-1
ANSI/IEEE 488-1978 Standard 15-8
AP Instruction ..... 10-2 thru 10-4, C-2
AR Instruction ..... 6-9 thru 6-13, C-2
AS Instruction 10-7, 10-8, C-2
<ASC> 14-4
ASCII Character Codes ..... A-2 thru A-10
Abort Device Control Instruction, ESC.J 16-39, C-24
Abort Graphics Instruction, ESC.K 14-25, 14-26, C-24
Absolute Character Size Instruction, SI ..... 7-14 thru 7-16, C-19
Absolute Coordinates 2-13, 2-14
Absolute Direction Instruction, DI ..... 7-21 thru 7-25, C-5
Absolute Plotting (Definition) 4-1
Absolute Plotting with PA, PU, and PD ..... 4-4 thru 4-6
Acceleration (Definition) 10-1
Acceleration Select Instruction, AS 10-7, 10-8, C-2
Acknowledgment String (Definition) 16-33
Acknowledgment String, Set in ESC.H or I Instruction 16-36
Adaptive Line Type ..... 5-6 thru 5-9
Addressable Mode (HP-IB) 15-2
Addressing Protocol (HP-IB) 15-3
Advance Page Instruction, AF or AH ..... 10-12 thru 10-14, C-1
Allocate Configurable Memory Instruction, ESC.T ..... 14-5 thru 14-10, C-26
Arc Absolute Instruction, AA ..... 6-9 thru 6-13, C-1
Arc Fonts, *See* Variable-Space Fonts
Arc Relative Instruction, AR ..... 6-9 thru 6-13, C-2
Auto Feed Key, *See also* Operation and Interconnection Manual 10-13
Automatic Pen Operations Instruction, AP ..... 10-2 thru 10-4, C-2
Axes, Labels and Ticks for (Line Chart Example) ..... 8-3 thru 8-5
Axes, Ticks for (XT, YT, and TL Instructions) ..... 5-1 thru 5-5

b

BASIC
- Description 1-6
- Graphics Statements and HP-GL Compared 1-9, 1-10
- Sending and Receiving Data (Examples) ..... 15-3 thru 15-7
- Statements Used in This Manual 1-7, 1-8

ManualsPlus.com

Subject Index (Continued)

b (Continued)

BF Instruction 10-15, 10-18, C-2
BL Instruction 7-38, 7-39, C-3
Bar Chart, Complete Program ..... 8-9 thru 8-13
Baud Function Key, *See also* Operation and Interconnection Manual 16-19
Binary and Decimal Conversions A-1
Bit States, Method for Checking ..... 12-5 thru 12-7
Bits, Determining Logic States from Decimal Equivalent Values 14-21
Block I/O Error Checking 14-17, 14-18
Block Size, *See* Data Block Size
Buffer (Definition) 14-2
Buffer Label Instruction, BL 7-38, 7-39, C-3
Buffer Overflow 3-8, 16-18
Buffer Plot Instruction, BF 10-15, 10-16, C-2
Buffered Labels
- LO Instruction and 7-3, 7-32, 7-39, 7-40
- Obtaining Information about ..... 7-40 thru 7-42
- Printing 7-3, 7-7, 7-39, 7-40

Buffers, *See also* Downloadable Character Buffer, Logical I/O Buffer, Physical I/O Buffer, Polygon Buffer, Replot Buffer, and Vector Buffer
- Allocating Memory ..... 3-18 thru 3-20, 14-5 thru 14-10
- Determining Current Size 14-11, 14-12, 14-15, 14-16

Bypass Function Key, *See also* Operation and Interconnection Manual 16-2

c

CA Instruction ..... 11-4 thru 11-11, C-3
CC Instruction ..... 11-26 thru 11-28, C-3
CD Line, Used in Hardwire Handshake 16-24
CHR$ Function ..... 7-5 thru 7-7, 11-5 thru 11-7, 11-19
CI Instruction ..... 6-5 thru 6-9, C-3
CM Instruction ..... 11-11, 11-14, 11-17, 11-19 thru 11-25, C-4
CP Instruction ..... 7-33 thru 7-36, C-4
CS Instruction ..... 11-4 thru 11-12, C-4
CT Instruction ..... 6-3 thru 6-5, C-4
CV Instruction ..... 10-10 thru 10-12, C-5
Cables (RS-232-C), *See also* Connector Pin Allocations and Operation and Interconnection Manual 16-19
Carousel, Determining Current Type 13-12, 13-13
Carriage-Return Character (CR), How to Send 7-6, 7-7

ManualsPlus.com

Subject Index (Continued)

C (Continued)

Carriage-Return Point 7-5, 7-6, 7-23, 7-24
Carriage-Return Point (Definition) 7-2
Cartesian Coordinate System 2-1
Character Chord Angle Instruction, CC ..... 11-26 thru 11-28, C-3
Character Origin 7-5, 7-6, 7-12, 7-13
Character Origin (Definition) 7-2
Character Plot Cell
  Average 7-12, 7-13
  Definition 7-2
  Description ..... 7-11 thru 7-14
  Line 7-12
  Space 7-12
Character Plot Instruction, CP ..... 7-33 thru 7-36, C-4
Characters, *See also* Labels
  Designing Your Own ..... 11-28 thru 11-41
  Direction 7-13
  Direction, Absolute ..... 7-21 thru 7-25
  Direction, Relative ..... 7-25 thru 7-30
  Height 7-12
  Position ..... 7-4 thru 7-6, 7-13, 7-14
  Primitive Grid ..... 7-12, 11-29 thru 11-41
  Size 7-13
  Size, Absolute ..... 7-14 thru 7-16
  Size, Relative ..... 7-16 thru 7-18
  Slant ..... 7-13, 7-18 thru 7-20
  Spacing 7-13, 7-36, 7-37
  Width 7-12
Character Selection Mode Instruction, CM ..... 7-3, 11-11, 11-14, 11-20 thru 11-22, C-4
Character Selection Modes
  Definition 11-2
  Fallback Mode 11-14
  HP 7-Bit Mode ..... 11-4, 11-12 thru 11-15
  HP 8-Bit Mode 11-12, 11-13, 11-14, 11-15, 11-16
  How to Choose 11-10
  How to Designate and Invoke Character Sets 11-11
  How to Send Characters in a Label Instruction 11-13
  ISO 7-Bit Mode ..... 11-13, 11-14, 11-16 thru 11-18
  ISO 8-Bit Mode ..... 11-13, 11-14, 11-18 thru 11-20
  In-Use Code Table ..... 11-11 thru 11-13
  Response to Control Characters 11-13, 11-14

ManualsPlus.com

Subject Index (Continued)

C (Continued)

Character Sets, *See also* Character Selection Modes
- All Characters Listed ..... A-2 thru A-10
- Definition 11-1
- Designating and Selecting Standard and Alternate ..... 11-4 thru 11-26
- Fixed-Space Fonts ..... 11-1, 11-2, A-5 thru A-7
- Fonts, Described 11-2, 11-3
- How to Send Characters in a Label Instruction ..... 11-5 thru 11-7
- Katakana8 ..... 11-15, 11-16
- Linked in HP 8-Bit Mode 11-15, 11-16
- Roman8 ..... 11-15, 11-16
- Selecting Standard and Alternate with Shift-In/Shift-Out 11-6, 11-7
- Variable-Space Fonts ..... 11-1, 11-2, A-8 thru A-10

Character Slant Instruction, SL ..... 7-18 thru 7-20, C-19
Chord Tolerance (Definition) 6-2
Chord Tolerance, Effects on Circle Smoothness 6-7, 6-8
Chord Tolerance Instruction, CT ..... 6-3 thru 6-5, C-4
Chord (Definition) 6-2
Circle Instruction, CI ..... 6-5 thru 6-9, C-3
Clipping ..... 2-9 thru 2-12
Clipping (Definition) 9-1
Compatibility with Other HP-GL Plotters 3-5, 3-6
Configurable Graphics Memory, *See* Buffers
Connector Pin Allocations ..... 16-11 thru 16-17
Constant (Definition) 4-2
Control Characters, Effect in Character Selection Modes 11-13, 11-14
Coordinate System
- Cartesian 2-1, 2-2
- Default Orientation of Plotter 2-3, 2-4
- Rotated ..... 9-2 thru 9-5

Coordinates, Absolute 2-13, 2-14
Coordinates, Relative 2-14, 2-15
Cross-Hatch (Definition) 6-3
Current Pen Position 2-13, 2-15
Current Pen Status 2-12, 2-13
Current Units (Definition) 4-1
Curved Line Generator ..... 10-10 thru 10-12, 14-7, 14-8
Curved Line Generator Instruction, CV ..... 10-10 thru 10-12, C-5

ManualsPlus.com

Subject Index (Continued)

d

DC Instruction 12-4, C-5
<DEC> 14-4
DF Instruction ..... 1-12, 3-8 thru 3-10, C-5
DI Instruction ..... 7-21 thru 7-25, C-5
DI and SI, Interactions of Parameters 7-42, 7-43
DI and SR, Interactions of Parameters ..... 7-44 thru 7-46
DL Instruction ..... 11-36 thru 11-41, C-5
DP Instruction 12-2, 12-3, C-6
DR Instruction ..... 7-25 thru 7-30, C-6
DR and SI, Interactions of Parameters 7-43, 7-44
DR and SR, Interactions of Parameters 7-46, 7-47
DS Instruction ..... 11-11, 11-12, 11-17, 11-22 thru 11-24, C-6
DT Instruction ..... 7-8 thru 7-10, C-6
DTR Line, Used in Hardwire Handshake 16-24
Data Block Size (Definition) 16-33
Data Block Size, Set in ESC.H or I Instruction ..... 16-35 thru 16-39
Data Terminal Ready (Definition) 16-33
Data Terminal Ready (DTR or CD) Line 16-24
Decimal and Binary Conversions A-1
Decimal Parameter Format 3-4
Default Conditions (Device-Control) 14-26
Default Conditions (HP-GL)
  Description 1-12
  Established by DF Instruction ..... 3-8 thru 3-10, A-11 thru A-13
  Established by IN Instruction 3-11, 3-12, A-13, A-14
Default (Definition) 3-2
Default Instruction, DF ..... 3-8 thru 3-10, C-5
Default P1 and P2 Coordinates 3-13, A-14
Default Values (Device-Control Syntax) 14-5
Define Downloadable Character Instruction, DL ..... 11-36 thru 11-41, C-5
Define Key Instruction, KY ..... 10-20 thru 10-23, C-11
Define Label Terminator Instruction, DT ..... 7-8 thru 7-10, C-6
Delimiters (Device-Control) 14-4, 14-5
Designate Alternate Character Set Instruction, CA ..... 11-4 thru 11-9, 11-12, C-3
Designate Character Set into Slot Instruction, DS ..... 11-11, 11-12, 11-17, 11-22 thru 11-24, C-6
Designate Standard Character Set Instruction, CS ..... 11-4 thru 11-8, 11-12, C-4
Developing Plots ..... 1-10, 1-11, 8-1 thru 8-16
Deviation Distance ..... 6-3 thru 6-5

ManualsPlus.com

Subject Index (Continued)

d (Continued)

Device-Control Instructions
- Description 1-5, 1-6, 14-2
- Errors ..... 14-16 thru 14-18, B-3, B-4
- General (HP-IB and RS-232-C) ..... 14-1 thru 14-27
- How to Send 14-3
- RS-232-C ..... 16-27 thru 16-31, 16-34 thru 16-40
- Syntax 14-4, 14-5

Digitize Clear Instruction, DC 12-4, C-5
Digitize Point Instruction, DP 12-2, 12-3, C-6
Digitizing
- Definition 12-1
- HP-IB Interrupts and Polling 12-7
- Manual Method 12-4, 12-5
- Monitoring the Status Byte ..... 12-5 thru 12-7
- Preparing the Plotter 12-1, 12-2
- Sight 12-2, 12-3

Direct Function Key, *See also* Operation and Interconnection Manual 14-13, 16-10, 16-20
Direct Mode (RS-232-C) 16-10, 16-20
Disconnection Modes (RS-232-C) 16-10
Display, Writing Messages on ..... 10-17 thru 10-20
Documentation for the HP 7550 iii
Download (Definition) 11-2
Downloadable Character Buffer ..... 3-18 thru 3-20, 11-39, 11-40, 14-7
Duplex Function Key and Monitor Modes, *See also* Operation and Interconnnection Manual 14-24, 14-25
Duplex Function Key and Remote/Local Modes, *See also* Operation and Interconnection Manual 16-5, 16-6, 16-20

e

E-Mask 13-5
EA Instruction ..... 6-27 thru 6-33, C-7
EP Instruction 6-26, 6-31, 6-36, 6-43, 6-44, C-7
ER Instruction ..... 6-33 thru 6-36, C-7
ES Instruction 7-36, 7-37, C-7
ESC.(Instruction 16-39, 16-40, C-26
ESC.) Instruction 16-40, C-26
ESC.@ Instruction ..... 14-12 thru 14-15, 14-17, 14-18, 16-34, C-22
ESC.A Instruction 14-27, C-23
ESC.B Instruction ..... 14-15, 14-16, 16-24 thru 16-26, 16-34, C-23

ManualsPlus.com

Subject Index (Continued)

e (Continued)

ESC.E Instruction ..... 14-16 thru 14-18, C-23
ESC.H Instruction ..... 16-34 thru 16-37, C-24
ESC.I Instruction ..... 16-34, 16-37 thru 16-39, C-24
ESC.J Instruction 16-39, C-24
ESC.K Instruction 14-25, 14-26, C-24
ESC.L Instruction 14-9, 14-11, 14-26, C-24
ESC.M Instruction ..... 16-28 thru 16-30, 16-34 thru 16-37, C-24
ESC.N Instruction ..... 16-28, 16-31, 16-34 thru 16-37, C-25
ESC.O Instruction ..... 14-9, 14-19 thru 14-22, C-25
ESC.P Instruction 16-27, 16-34, 16-35, C-25
ESC.Q Instruction ..... 14-22 thru 14-25, C-25
ESC.R Instruction 14-26, C-25
ESC.S Instruction 14-11, 14-12, C-26
ESC.T Instruction *See also* GM Instruction ..... 14-5 thru 14-10, C-26
ESC.U Instruction 10-22, 14-27, C-26
ESC.Y Instruction 16-39, 16-40, C-26
ESC.Z Instruction 16-40, C-26
ESC Character in Device-Control Instructions ..... 14-2 thru 14-4
EW Instruction ..... 6-21 thru 6-27, C-8
Eavesdrop Configuration
  General 16-3, 16-4
  Local Mode 16-8, 16-9
  Remote Mode 16-6, 16-7
Eavesdrop Function Key, *See also* Operation and
  Interconnection Manual 16-4
Echo Terminate Character (Definition) 16-32
Echo Terminate Character, Set in ESC.M Instruction 16-28
Echoing Data, *See* Duplex Function Key
Edge Polygon Instruction, EP 6-26, 6-31, 6-36, 6-43, 6-44, C-7
Edge Rectangle Absolute Instruction, EA ..... 6-27 thru 6-33, C-7
Edge Rectangle Relative Instruction, ER ..... 6-33 thru 6-36, C-7
Edge Wedge Instruction, EW ..... 6-21 thru 6-27, C-8
End Flush Mode Instruction, ESC.U 10-22, 14-27, C-26
Enhanced Function Key, *See also* Operation and
  Interconnection Manual
  Effect on DL Instruction 11-38
  Effect on IW Instruction 9-6
  Effect on OD Instruction 12-3
  Effect on OW Instruction 9-9
  Effect on UC Instruction ..... 11-29 thru 11-34
Enlarging/Reducing Plots ..... 9-13 thru 9-16

ManualsPlus.com

Subject Index (Continued)

e (Continued)

Enquire/Acknowledge Handshake ..... 16-21, 16-23, 16-24, 16-27, 16-31 thru 16-39
Enquiry Character (Definition) 16-32, 16-33
Enquiry Character, Set in ESC . H or I Instruction ... 16-35 thru 16-39
ENTER Statement 1-8, 13-2
Error Checking, Block I/O ..... 14-14, 14-16 thru 14-18
Errors
  Determining Device-Control 14-16, 14-17
  Determining HP-GL 13-8, 13-9
  RS-232-C Transmission 16-18
  Table of Device-Control 14-16, 14-17, B-3, B-4
  Table of HP-GL ..... 3-6 thru 3-8, B-1, B-2
Escape Function and Flush Mode 10-18, 10-22, 10-26
Escape Function Key, *See also* Operation and Interconnection Manual 10-26
Examples, How to Use ..... 1-6 thru 1-9
Execute (Definition) 14-2
Extra Space Instruction, ES 7-36, 7-37, C-7

f

FORTRAN (Description) 1-6
FORTRAN, Sending and Receiving Data (HP 9000, Series 500 Computer) 15-5, 15-7, 15-8
FP Instruction ..... 6-44 thru 6-48, C-8
FS Instruction 10-6, 10-7, C-8
FT Instruction ..... 6-13 thru 6-16, C-8
Fallback Mode 11-14
Fill Polygon Instruction, FP ..... 6-44 thru 6-48, C-8
Fill Rectangle Absolute Instruction, RA ..... 6-27 thru 6-33, C-17
Fill Rectangle Relative Instruction, RR ..... 6-33 thru 6-36, C-17
Fill Type Instruction, FT ..... 6-13 thru 6-16, C-8
Fill Wedge Instruction, WG ..... 6-21 thru 6-27, C-22
Fixed Line Type ..... 5-6 thru 5-9
Fixed-Space Fonts
  All Characters Listed ..... A-5 thru A-7
  Definition 7-1
  Description ..... 11-2 thru 11-4
Flush Mode 10-18, 10-22, 10-26, 14-27
Force (Definition) 10-1
Force, Default for Each Carousel 10-7
Force Select Instruction, FS 10-6, 10-7, C-8

ManualsPlus.com

Subject Index (Continued)

f (Continued)

Front-Panel Display, Writing Messages on ..... 10-17 thru 10-20
Function Keys, Redefining 10-17, 10-18, 10-20, 10-21

g

GC Instruction 10-25, C-9
GM Instruction ..... 3-17 thru 3-19
GP Instruction 10-4, 10-5, C-9
Graphics Limits
  Determining Hard-Clip Limits 9-10, 9-11
  Determining Soft-Clip Limits 9-9, 9-10
  Effects on Plotting Instructions ..... 4-9 thru 4-11
  Hard-Clip, Description 2-9, 2-10
  Lost Mode 4-11
  Setting Soft-Clip Limits ..... 9-5 thru 9-9
  Soft-Clip (Window), Description ..... 2-10 thru 2-12
Graphics Memory, *See* Buffers
Graphics Memory Instruction, GM ..... 3-18 thru 3-20
Graphics Sets GL and GR, General In-Use
  Code Table ..... 11-11 thru 11-13
Graphics Sets GL and GR, Implementation in
  Each Character Selection Mode ..... 11-15 thru 11-19
Graphics Slots G0 through G3, General In-Use
  Code Table ..... 11-11 thru 11-13
Graphics Slots G0 through G3, Implementation in
  Each Character Selection Mode ..... 11-15 thru 11-19
Grid (Definition) 5-1
Grid, Plotting Example 5-4, 5-5
Group Count Instruction, GC 10-25, C-9
Group Pen Instruction, GP 10-4, 10-5, C-9

h

HP 150 Computer 15-4, 15-6
HP 3000 Computer 1-9
HP 7-Bit Character Selection Mode ..... 11-4, 11-10, 11-13 thru 11-15
HP 8-Bit Character Selection Mode 11-4, 11-10, 11-13, 11-15, 11-16
HP 9000, Series 200 Computers 1-9, 15-4, 15-6
HP 9000, Series 500 Computers 1-9, 15-4, 15-5, 15-7
HP 9816, 9826, and 9836 Computers 1-9, 15-4, 15-6
HP-GL
  BASIC Graphics Statements Compared with 1-9, 1-10
  Description 1-5

ManualsPlus.com

Subject Index (Continued)

h (Continued)

Errors, Determining 13-8, 13-9
Errors, Listed ..... 3-6 thru 3-8, B-1, B-2
How to Send to Plotter ..... 1-10 thru 1-12
Instruction Set, Alphabetical List ..... 1-2 thru 1-4, C-1 thru C-22
Strings 1-8
Syntax ..... 3-2 thru 3-6
HP-IB
Addressing (General) 15-2, 15-3
Addressing Sequences and Codes ..... 15-11 thru 15-13
Bus Capabilities ..... 15-14 thru 15-16
Definition 15-1
Interface Functions ..... 15-13 thru 15-14
Interfacing ..... 15-1 thru 15-18
Interrupts 12-7
Lines ..... 15-8 thru 15-10
Operations (Messages and Commands) 15-10, 15-11
Serial and Parallel Polling 12-7, 15-17, 15-18
HP-IB Function Key, *See also* Operation and
Interconnection Manual 15-2
Handshake Function Key, *See also* Operation and
Interconnection Manual 14-13, 16-20, 16-21
Handshake Methods (RS-232-C)
Device-Control Instructions and Parameters Used 16-34, 16-35
Enquire/Acknowledge ..... 16-21, 16-23, 16-24, 16-27, 16-32 thru 16-39
General 16-20, 16-21, 16-26, 16-27, 16-31, 16-34, 16-35
Hardwire 14-14, 16-11, 16-21, 16-24, 16-26, 16-27, 16-34
Predefined 16-26, 16-27
Software (Buffer) Checking ..... 16-21, 16-24 thru 16-26, 16-34
Tailoring Your Own ..... 16-31 thru 16-35
Xon-Xoff ..... 16-21 thru 16-23, 16-27, 16-31, 16-34, 16-37 thru 16-39
Handshake Parameters (Definitions) ..... 16-31 thru 16-34
Handshake Parameters, Set by Front-Panel or
ESC.P Instruction 16-27
Hard-Clip Limits
Definition 9-1
Description 2-9, 2-10
Determining Current Location 9-10, 9-11
Listed for Each Paper Size 2-9, 9-11
Rotated ..... 9-2 thru 9-5
Hardwire Handshake 14-14, 16-11, 16-21, 16-24, 16-26, 16-27, 16-34
Hatch (Definition) 6-3
Hewlett-Packard Interface Bus, *See also* HP-IB ..... 15-1 thru 15-18

ManualsPlus.com

Subject Index (Continued)

I

I/O Buffer, *See also* Physical I/O Buffer and
Logical I/O Buffer 14-2, 14-6
I/O Buffer (Definition) 14-2
I/O Errors ..... 14-16 thru 14-18
IBM Personal Computer 1-9
IM Instruction ..... 13-4 thru 13-7, C-9
Used for Digitizing (HP-IB) 12-7
Used for Serial and Parallel Polling (HP-IB) 15-17, 15-18
IN Instruction 1-12, 3-11, 3-12, C-10
IP Instruction ..... 3-12 thru 3-14, C-10
ISO 2022.2 Standard 11-14
ISO 7-Bit Character Selection Mode ..... 11-10, 11-13, 11-16 thru 11-18
ISO 8-Bit Character Selection Mode ..... 11-10, 11-13, 11-18 thru 11-20
IV Instruction ..... 11-11, 11-12, 11-24 thru 11-26, C-10
IW Instruction ..... 9-5 thru 9-9, C-11
Immediate Response String (Definition) 16-33
Immediate Response String, Set in ESC.N Instruction 16-31
In-Use Code Table
General for All Character Selection Modes ..... 11-11 thru 11-13
HP 7-Bit Mode 11-15
HP 8-Bit Mode 11-15
ISO 7-Bit Mode 11-16, 11-17
ISO 8-Bit Mode 11-18
Increments, Relative 2-14
Initialize (Definition) 3-2
Initialize Instruction, IN 3-11, 3-12, C-10
Initialized Conditions (HP-GL) 1-12, 3-11, 3-12, A-13, A-14
Input Mask Instruction, IM ..... 13-4 thru 13-7, C-9
Used for Digitizing (HP-IB) 12-7
Used for Serial and Parallel Polling (HP-IB) 15-17, 15-18
Input P1 and P2 Instruction, IP ..... 3-12 thru 3-14, C-10
Input Window Instruction, IW ..... 9-5 thru 9-9, C-10
Integer Parameter Format 3-4
Interactive Programming (Definition) 10-2
Interactive Programs 10-17, 10-18
Intercharacter Delay (Definition) 16-32
Intercharacter Delay, Set in ESC.N Instruction 16-31
Interfacing
HP-IB ..... 15-1 thru 15-18
RS-232-C/CCITT V.24 ..... 16-1 thru 16-40
Invoke Character Slot Instruction, IV ..... 11-11, 11-12, 11-24 thru 11-26, C-10

ManualsPlus.com

Subject Index (Continued)

k

KY Instruction ..... 10-20 thru 10-23, C-11
Katakana8 Character Set 11-15, 11-16
Keyboard Mode (Definition) 10-2
Keyboard Mode (WD Instruction) ..... 10-18 thru 10-20

l

LB Instruction ..... 7-2 thru 7-8, C-11
LO Instruction ..... 7-30 thru 7-33, C-11
LT Instruction ..... 5-6 thru 5-9, C-11
Label Buffer, *See also* Buffered Labels ..... 7-3, 7-7, 7-38 thru 7-40
Label Instruction, LB ..... 7-2 thru 7-8, C-11
Label Origin Instruction, LO ..... 7-30 thru 7-33, C-11
Label Terminator
  Changing with the DT Instruction ..... 7-8 thru 7-10
  Definition 7-2
  How to Send 7-6, 7-7
  Syntax Notation 3-3
Labeling with Alternate Character Sets, *See* Character Sets
Labeling with Variables 7-10, 7-11
Labels, *See also* Characters
  Absolute Direction ..... 7-21 thru 7-25
  Determining the Length ..... 7-40 thru 7-42
  Direction Affected by P1, P2, and
    Negative Parameters ..... 7-42 thru 7-47
  Mirror Images ..... 7-15, 7-16, 7-42 thru 7-47
  Parameter Formats 3-4
  Position ..... 7-4 thru 7-6, 7-30 thru 7-33
  Relative Direction ..... 7-25 thru 7-30
  Rotating the Direction ..... 7-23 thru 7-25
Leased-Line Disconnection 16-10
Line (Definition for Labeling) 7-2
Line Chart, Complete Program ..... 8-2 thru 8-9
Line Chart with Variable Parameters 4-13, 4-14
Line-Feed Character (LF), How to Send 7-6, 7-7
Line Type Instruction, LT ..... 5-6 thru 5-9, C-11
Line Types
  Adaptive ..... 5-7 thru 5-9
  Effects on Circles 6-8, 6-9
  Effects on Fill Patterns 6-15, 6-16
  Fixed ..... 5-7 thru 5-9

ManualsPlus.com

Subject Index (Continued)

l (Continued)

Listen-Only Mode (HP-IB) 15-2, 15-3
Literal String (Definition) 3-1
Loading Paper with the PG, AF, or
AH Instructions ..... 10-12 thru 10-14
Local Mode (RS-232-C) 16-5, 16-6, 16-8, 16-9, 16-19
Logical I/O Buffer
Allocating Memory ..... 14-6, 14-9, 14-12 thru 14-15
Definition 14-2
Determining Size of 14-11, 14-12, 14-15, 14-16
Used in Enquire/Acknowledge Handshake 16-23, 16-24
Used in Hardwire Handshake 16-24
Used in Software Checking Handshake ..... 16-24 thru 16-26
Used in Xon-Xoff Handshake 16-22, 16-23
Lost Mode 4-11

m

Maximum Logical I/O Buffer Size (Definition) 16-33
Media sizes 1-1
Media types, *See also* Operation and
Interconnection Manual 1-1
Memory, *See* Buffers
Mirror Images of Labels ..... 7-15, 7-16, 7-42 thru 7-47
Mirror Images of Plotted Data 9-17, 9-18
Mnemonic, Definition of 3-1
Modem Function Key, *See also* Operation and
Interconnection Manual 14-13, 16-10, 16-20
Modem Mode (RS-232-C) 16-10, 16-20
Monitor Modes ..... 14-14, 14-22 thru 14-25

n

NOP Instructions A-14, A-15
NR Instruction 10-14, 10-15, C-12
No Operation (NOP) Instructions A-14, A-15
Not-Ready Instruction, NR 10-14, 10-15, C-12
Not-Ready State 10-13, 10-14, 10-15

o

OA Instruction 13-7, 13-8, C-12
OC Instruction 13-8, C-13
OD Instruction 12-3, C-13

ManualsPlus.com

Subject Index (Continued)

O (Continued)

OE Instruction 13-8, 13-9, C-13
OF Instruction 13-9, C-13
OG Instruction 10-22, 10-26, 10-27, C-13
OH Instruction 9-10, 9-11, C-13
OI Instruction 13-10, C-14
OK Instruction ..... 10-23 thru 10-25, C-14
OL Instruction ..... 7-40 thru 7-42, C-14
OO Instruction 13-10, C-14
OP Instruction ..... 9-11 thru 9-12, C-14
OS Instruction 13-11, 13-12, C-14
OT Instruction 13-12, 13-13, C-15
OW Instruction 9-9, 9-10, C-15
Omitting Parameters (HP-GL) 3-3
Omitting Parameters (Device Control) 14-5
Organization Chart, Program Using
Absolute Coordinates 6-32, 6-33
Organization Chart, Program Using Relative Coordinates 6-36
Origin of Coordinate System 2-1
Output, Controlling in RS-232-C Interfacing ..... 16-28 thru 16-31
Output Actual Position and Pen Status Instruction, OA 13-7, 13-8, C-12
Output Buffer Size When Empty Instruction, ESC.L 14-9, 14-11, 14-26, C-24
Output Buffer Space Instruction, ESC.B ..... 14-15, 14-16, 16-24 thru 16-26, 16-34, C-23
Output Carousel Type Instruction, OT ..... 13-12 thru 13-14, C-15
Output Commanded Position and
Pen Status Instruction OC 13-8, C-13
Output Configurable Memory Size Instruction, ESC.S 14-11, 14-12, C-26
Output Digitized Point and Pen Status Instruction, OD 12-3, C-13
Output Error Instruction, OE 13-8, 13-9, C-13
Output Extended Error Instruction, ESC.E .... 14-16 thru 14-18, C-23
Output Extended Status Instruction, ESC.O ..... 14-9, 14-19 thru 14-22, C-25
Output Factors Instruction, OF 13-9, C-13
Output Group Count Instruction, OG ..... 10-22, 10-26 thru 10-29, C-13
Output Hard-Clip Limits Instruction, OH 9-10, 9-11, C-13
Output Identification Instruction, ESC.A 14-27, C-23
Output Identification Instruction, OI 13-10, C-14
Output Initiator Character (Definition) 16-32
Output Initiator Character, Set in ESC.M Instruction 16-29

ManualsPlus.com

Subject Index (Continued)

O (Continued)

Output Key Instruction, OK ..... 10-23 thru 10-25, C-14
Output Label Length Instruction, OL ..... 7-40 thru 7-42, C-14
Output Options Instruction, OO 13-10, 13-11, C-14
Output P1 and P2 Instruction, OP ..... 9-11 thru 9-12, C-14
Output Response Terminator (Device-Control) 14-5
Output Response Terminator (HP-GL and Device-Control) 3-5
Output Responses
  Effects of Handshake Parameters on 16-37
  Hints for Obtaining ..... 13-2 thru 13-4
  HP-IB Notes 13-3
  RS-232-C Notes 13-4
  Summary of Number and Type of Parameters 13-13, 13-14
OUTPUT Statement 1-8
Output Status Instruction, OS 13-11, 13-12, C-14
Output Terminator (Definition) 16-32
Output Terminator, Set in ESC.M Instruction 16-29
Output Trigger Character (Definition) 16-31
Output Trigger Character, Set in ESC.M Instruction 16-28
Output Window Instruction, OW 9-9, 9-10, C-15

P

P-Mask 13-6, 13-7
P1 and P2
  Default 3-13, A-14
  Description ..... 2-5 thru 2-9
  Determining Current Location 9-11, 9-12
  Effects on Label Direction ..... 7-42 thru 7-47
  Effects on Relative Character Size ..... 7-16 thru 7-18
  Effects on Relative Label Direction ..... 7-25 thru 7-30
  Locations after Coordinate System Rotation ..... 3-13, 9-2 thru 9-4
  Preparing Equal-Sized Plots on One Page 9-12, 9-13
  Setting Locations of ..... 3-12 thru 3-17
  Used for Reducing/Enlarging Plots ..... 9-13 thru 9-16
  Used with Windows When Scaling Is On ..... 9-14 thru 9-16
PA Instruction ..... 4-4 thru 4-6, C-15
PB Instruction 7-3, 7-7, 7-8, 7-38, 7-39, 7-40, C-15
PD Instruction ..... 4-3 thru 4-6, 4-8, C-15
PG Instruction ..... 10-12 thru 10-14, C-16
PM Instruction ..... 6-37 thru 6-43, C-16
PR Instruction ..... 4-6 thru 4-9, C-16
PT Instruction 6-20, 6-21, C-16

ManualsPlus.com

Subject Index (Continued)

P (Continued)

PU Instruction ..... 4-3 thru 4-6, 4-8, C-17
Page Feed Instruction, PG ..... 10-12 thru 10-14, C-16
Paper Feed, Determining Current Mode 13-10
Paper Size, Default P1 and P2 for 3-13, A-14
Paper Size, Plotting Range for 2-9, 9-11
Parallel Poll (HP-IB) 13-4, 13-5, 13-6, 13-7, 15-10, 15-17, 15-18
Parameter
  Definition 3-1
  Formats (Integer, Decimal, Label) 3-4
  Optional (Device-Control) 14-4, 14-5
  Optional (HP-GL) 3-3
Parity Function Key, *See also* Operation and
  Interconnection Manual 11-10, 16-19
Parse (Definition) 14-2
Parse Mode Function Key, *See also* Operation and
  Interconnection Manual 14-14, 14-24
Parse Monitor Mode ..... 14-14, 14-22 thru 14-25
Pascal, Description 1-6
Pen-Dn Function Key, *See also* Operation and
  Interconnection Manual 4-4
Pen Down Instruction, PD 4-3, 4-4, 4-6, 4-8, C-15
Pen Position
  Description 2-12, 2-13, 2-15
  Determining Actual 13-7, 13-8
  Determining Commanded 13-8
Pen Stalls, Determining Which Are Occupied 13-12, 13-13
Pen Status
  Description 2-12, 2-13
  Determining Actual 13-7, 13-8
  Determining Commanded 13-8
Pen Thickness Instruction, PT 6-20, 6-21, C-16
Pen-Up Function Key, *See also* Operation and
  Interconnection Manual 4-4
Pen Up Instruction, PU ..... 4-3 thru 4-6, 4-8, C-17
Pen Types, *See also* Operation and
  Interconnection Manual 1-1
Pens, Grouping ..... 10-4 thru 10-6
Physical I/O Buffer
  Allocating Memory 14-6
  Definition 14-2
  Determining Size of 14-11, 14-12

ManualsPlus.com

Subject Index (Continued)

p (Continued)

Pie Chart, Complete Program ..... 8-13 thru 8-16
Pie Chart, Simple 1-11
Pin Allocations in Connectors ..... 16-11 thru 16-17
Plot Absolute Instruction, PA ..... 4-4 thru 4-6, C-15
Plot Relative Instruction, PR ..... 4-6 thru 4-9, C-16
Plot Size, Reducing/Enlarging ..... 9-13 thru 9-16
Plotter Coordinate System Orientation 2-3, 2-4
PLOTTER IS Statement 2-10
Plotter Units (Definition) 3-2
Plotter Units, Description and Range ..... 2-2 thru 2-4
Plotter-Off Instruction, ESC.Z or ESC.) 16-40, C-26
Plotter-On Instruction, ESC.Y or ESC.(..... 16-39, 16-40, C-26
Plotting Area, Placing More Than One Plot on 9-12, 9-13
Plotting Range for Each Paper Size 2-9
Plotting with Variables ..... 4-12 thru 4-14
Point 2-1, 2-2
Poll (Definition) 10-2
Polling, HP-IB Serial and Parallel 15-17, 15-18
Polling the Plotter for a Successful
  Paper Feed ..... 10-13, 14-19, thru 14-22
Polygon Buffer ..... 6-48 thru 6-51, 14-7
Polygon (Definition) 6-2
Polygon Mode Instruction, PM ..... 6-37 thru 6-43, C-16
Polygons, How to Define 6-38, 6-39
Positioning Labels ..... 7-30 thru 7-36
Primitive Grid
  Character Plot Cell 7-12
  Downloadable Characters 11-38, 11-41
  User-Defined Characters ..... 11-28 thru 11-33
Print Buffered Label Instruction, PB 7-3, 7-7, 7-39, 7-40, C-15
PRINT #1 Statement 1-8
Programmed-Off (Definition) 16-2
Programmed-Off State, Set by ESC.Z or ESC.) 16-40
Programmed-On (Definition) 16-2
Programmed-On State, Set by ESC.Y or ESC.(..... 16-39, 16-40
Programming, Interactive 10-2, 10-17, 10-18
Programming Languages
  BASIC 1-6
  Device-Control Instructions 1-5, 1-6
  FORTRAN 1-6
  HP-GL 1-5
  Pascal 1-6

ManualsPlus.com

Subject Index (Continued)

p (Continued)

Programs for Line, Bar, and Pie Charts ..... 8-1 through 8-16
Proportional Fonts, *See* Variable-Space Fonts

r

RA Instruction ..... 6-27 thru 6-33, C-17
RO Instruction ..... 9-2 thru 9-5, C-17
RP Instruction 10-16, 10-17, C-17
RR Instruction ..... 6-33 thru 6-36, C-17
RS-232-C/CCITT V.24 Interface Implementation ..... 16-11 thru 16-15
RS-232-C/CCITT V.24 Interfacing, *See also*
Handshake Methods ..... 16-1 thru 16-40
RS-422-A Interface Implementation 16-11, 16-16, 16-17
Range of Plotter Units and User Units 2-4, 2-5
READ Statement 1-8, 13-2
READLN Statement 1-8, 13-2
Receive Mode Function Key, *See also* Operation and
Interconnection Manual 14-14
Receive Monitor Mode ..... 14-14, 14-22 thru 14-25
Receiving Data from the Plotter (HP-IB) ..... 15-6 thru 15-8
Record Size, *See* Data Block Size
Rectangle Instructions (RA, EA, RR, and ER) ..... 6-27 thru 6-36
Reducing/Enlarging Plots ..... 9-13 thru 9-16
Relative Character Size Instruction, SR ..... 7-16 thru 7-18, C-20
Relative Coordinates 2-14, 2-15
Relative Direction Instruction, DR ..... 7-25 thru 7-30, C-6
Relative Plotting (Definition) 4-1
Relative Plotting with PR, PU, and PD ..... 4-6 thru 4-9
Remote Mode (RS-232-C) ..... 16-5 thru 16-7, 16-9, 16-19
Replot Buffer 10-15, 14-7
Replot Function Key, *See also* Operation and
Interconnection Manual 10-16
Replot Instruction, RP 10-16, 10-17, C-17
Reset Function Key, *See also* Operation and
Interconnection Manual 3-11
Reset Instruction, ESC . R 14-26, C-25
Roman8 Character Set 11-15, 11-16
Rotate Coordinate System Instruction, RO ..... 9-2 thru 9-5, C-17
Rotate Function Key, *See also* Operation and
Interconnection Manual ..... 9-2 thru 9-5
Rotating Label Direction ..... 7-23 thru 7-25

ManualsPlus.com

Subject Index (Continued)

r (Continued)

Rotation of Coordinate System and P1 and P2 ..... 3-13, 9-2 thru 9-5
Rotation and Windows 9-6

S

S-Mask 13-6
SA Instruction ..... 11-5, 11-6, 11-9 thru 11-12, C-18
SC Instruction ..... 3-14 thru 3-17, A-18, C-18
SG Instruction 10-6, C-18
SI Instruction ..... 7-14 thru 7-16, C-19
SI and DI, Interactions of Parameters 7-42, 7-43
SI and DR, Interactions of Parameters 7-43, 7-44
SL Instruction ..... 7-18 thru 7-20, C-19
SM Instruction 5-5, 5-6, C-19
SP Instruction 4-2, 4-3, C-20
SR Instruction ..... 7-16 thru 7-18, C-20
SR and DI, Interactions of Parameters ..... 7-44 thru 7-46
SR and DR, Interactions of Parameters 7-46, 7-47
SS Instruction 11-5, 11-6, 11-9, 11-12, C-20
Scale Instruction, SC ..... 3-14 thru 3-17, C-18
Scaling ..... 2-5 thru 2-9, 3-14 thru 3-17
  Isotropic and Anisotropic 3-15, 3-16,
  Line Chart, Bar Chart, and Pie Chart Examples 8-2, 8-9, 8-13
  Reducing/Enlarging Plots ..... 9-13 thru 9-16
  Without Using the SC Instruction ..... A-15 thru A-17
Scaling Points (Definition) 3-2
Scaling Points, *See* P1 and P2
Select Alternate Character Set Instruction, SA ..... 11-5, 11-6, 11-9 thru 11-12, C-18
Select Pen Instruction, SP 4-2, 4-3, C-20
Select Pen Group Instruction, SG 10-6, C-18
Select Standard Character Set Instruction, SS 11-5, 11-6, 11-9, 11-12, C-20
Semilog Fill Type 6-19, 6-20
Sending Data to the Plotter (HP-IB) ..... 15-3 thru 15-5
Separator (Definition, HP-GL) 3-2
Separators, Optional and Required (HP-GL) 3-3
Serial Mode (Definition) 16-3
Serial Poll (HP-IB) 12-7, 13-6, 15-17
Service Request (HP-IB) 13-4, 13-6, 15-17
Set Extended Output and Handshake Mode
  Instruction, ESC.N ..... 16-31, 16-34 thru 16-37, C-25

ManualsPlus.com

Subject Index (Continued)

S (Continued)

Set Handshake Mode Instruction, ESC.P 16-27, 16-34, 16-35, C-25
Set Handshake Mode 1 Instruction, ESC.H .... 16-34 thru 16-37, C-24
Set Handshake Mode 2 Instruction, ESC.I ..... 16-34, 16-37 thru 16-39, C-24
Set Monitor Mode Instruction, ESC.Q ..... 14-22 thru 14-25, C-25
Set Output Mode Instruction, ESC.M ..... 16-28 thru 16-30, 16-34 thru 16-37, C-24
Set Plotter Configuration Instruction, ESC.@ ..... 14-12 thru 14-15, 14-18, 16-34, C-22
Shift-In, Used for Selecting Standard Character Set 11-6, 11-7
Shift-Out, Used for Selecting Alternate Character Set 11-6, 11-7
Single-Shift, Used in ISO 8-Bit Character Selection Mode 11-19
Soft-Clip Limits (Windows)
  Definition 9-1
  Description ..... 2-10 thru 2-12
  Determining Current Location 9-9, 9-10
  Setting Location of ..... 9-5 thru 9-9
Softkey Function Key, *See also* Operation and Interconnection Manual 10-19
Software (Buffer) Checking Handshake ..... 16-21, 16-24 thru 16-26, 16-34
Space (Definition for Labeling) 7-2
Speed, Pen
  Changing on Front Panel, *See* Operation and Interconnection Manual
  Changing with the VS Instruction ..... 10-8 thru 10-10
  Default for Each Carousel 10-9
  Definition 10-2
Spooling (Definition) 10-2
Spooling and Flush Mode 10-21, 10-22, 10-25, 10-26, 14-27
Stand-Alone Configuration
  General 16-4, 16-5
  Local Mode 16-9
  Remote Mode 16-9
Standalone Function Key, *See also* Operation and Interconnection Manual 16-4
Standard Function Key, *See also* Operation and Interconnection Manual
  Effect on DL Instruction 11-38
  Effect on IW Instruction 9-6
  Effect on OD Instruction 12-3
  Effect on OW Instruction 9-9

ManualsPlus.com

Subject Index (Continued)

S (Continued)

Effect on UC Instruction ..... 11-29 thru 11-34
Standby Mode (RS-232-C) 16-6, 16-19
Status Byte, Checking Bit States in ..... 12-5 thru 12-7
Status Byte, Determining Bit States of 14-21
Status of Plotter, Determining ..... 13-11, 13-12, 14-19 thru 14-22
Stop Bits 16-19
String (Definition) 3-1
Subpolygon (Definition) 6-2
Surface Chart, Program for 6-45, 6-46
Switched/Datex-Line Disconnection 16-10
Symbol Mode Instruction, SM 5-5, 5-6, C-19
Syntax
Compatibility with Other HP-GL Plotters 3-5, 3-6
Definition 3-1
Device-Control 14-4, 14-5
HP-GL ..... 3-2 thru 3-6

t

[TERM] (HP-IB) 3-5, 13-3
[TERM] (RS-232-C) 3-5, 13-4
TL Instruction ..... 5-2 thru 5-5, C-20
Terminator
Definition 3-2
Device-Control Instructions 14-5
HP-GL 3-2, 3-5
Label 3-3
Output Response ..... 3-5, 13-3, 13-4, 14-5, 16-28 thru 16-30
Tick (Definition) 5-1
Tick Length Instruction, TL ..... 5-2 thru 5-5, C-20
Ticks, Plotted Examples ..... 5-4, 5-5, 8-3 thru 8-5
Transmission Errors (RS-232-C) 16-18
Turnaround Delay (Definition) 16-32
Turnaround Delay, Set in ESC.M Instruction 16-28

u

UC Instruction ..... 11-28 thru 11-36, C-21
UF Instruction ..... 6-17 thru 6-20, C-21
Unloading Paper with the NR Instruction 10-14, 10-15
User Units
Converting to Plotter Units ..... A-15 thru A-17

ManualsPlus.com

Subject Index (Continued)

U (Continued)

Definition 3-2
Description and Range ..... 2-2, 2-4 thru 2-9
Specifying with the SC Instruction ..... 3-14 thru 3-17
User-Defined Character Instruction, UC ..... 11-28 thru 11-36, C-21
User-Defined Fill Type Instruction, UF ..... 6-17 thru 6-20, C-21

V

VS Instruction ..... 10-8 thru 10-10, C-21
Variable (Definition) 4-2
Variable Parameters in HP-GL Instructions
Examples for Various Computers, *See* Operation and Interconnection Manual
Examples for Various HP-IB Computers ..... 15-3 thru 15-5
How to Send to the Plotter ..... 4-12 thru 4-14, 7-10, 7-11
Line Chart Examples ..... 4-13, 4-14, 8-2 thru 8-9
Variable-Space Fonts
All Character Sets Listed ..... A-8 thru A-10
Changing Arc Smoothness ..... 11-26 thru 11-28
Definition 7-1, 7-2
Description ..... 11-2 thru 11-4
Vector Buffer ..... 10-10 thru 10-12, 14-7, 14-8
Vector Fonts, *See* Fixed-Space Fonts
Velocity, Pen
Changing on Front Panel, *See* Operation and Interconnection Manual
Changing with VS Instruction ..... 10-8 thru 10-10
Default for Each Carousel 10-9
Definition 10-2
Velocity Select Instruction, VS ..... 10-8 thru 10-10, C-21
Vertex (Definition) 6-2

W

WD Instruction ..... 10-18 thru 10-20, C-22
WG Instruction ..... 6-21 thru 6-27, C-22
Windows
Definition 9-1
Description ..... 2-10 thru 2-12
Determining Current Location 9-9, 9-10
Plotting Inside and Outside of 4-10, 4-11
Specifying Location of ..... 9-5 thru 9-9

ManualsPlus.com

Subject Index (Continued)

W (Continued)

Used for Reducing/Enlarging Portions of Plots ..... 9-14 thru 9-16
Write to Display Instruction, WD ..... 10-18 thru 10-20, C-22
WRITE Statement 1-8
WRITELN Statement 1-8

X

X-Axis 2-1, 2-2
X-Coordinate 2-1, 2-2
X-Tick Instruction, XT 5-1, 5-2, C-22
XT Instruction 5-1, 5-2, C-22
X,Y Coordinate Pair 2-1, 2-2
Xoff Threshold Level (Definition) 16-33
Xoff Threshold Level, Set in ESC.I Instruction ..... 16-37 thru 16-39
Xoff Trigger Character (Definition) 16-34
Xoff Trigger Character, Set in ESC.N Instruction 16-31
Xon Trigger Character (Definition) 16-34
Xon Trigger Character, Set in ESC.I Instruction 16-38
Xon-Xoff Handshake .... 16-21 thru 16-23, 16-34, 16-35, 16-37 thru 16-39

Y

Y-Axis 2-1, 2-2
Y-Coordinate 2-1, 2-2
Y-Tick Instruction, YT 5-1, 5-2, C-22
YT Instruction 5-1, 5-2, C-22

ManualsPlus.com