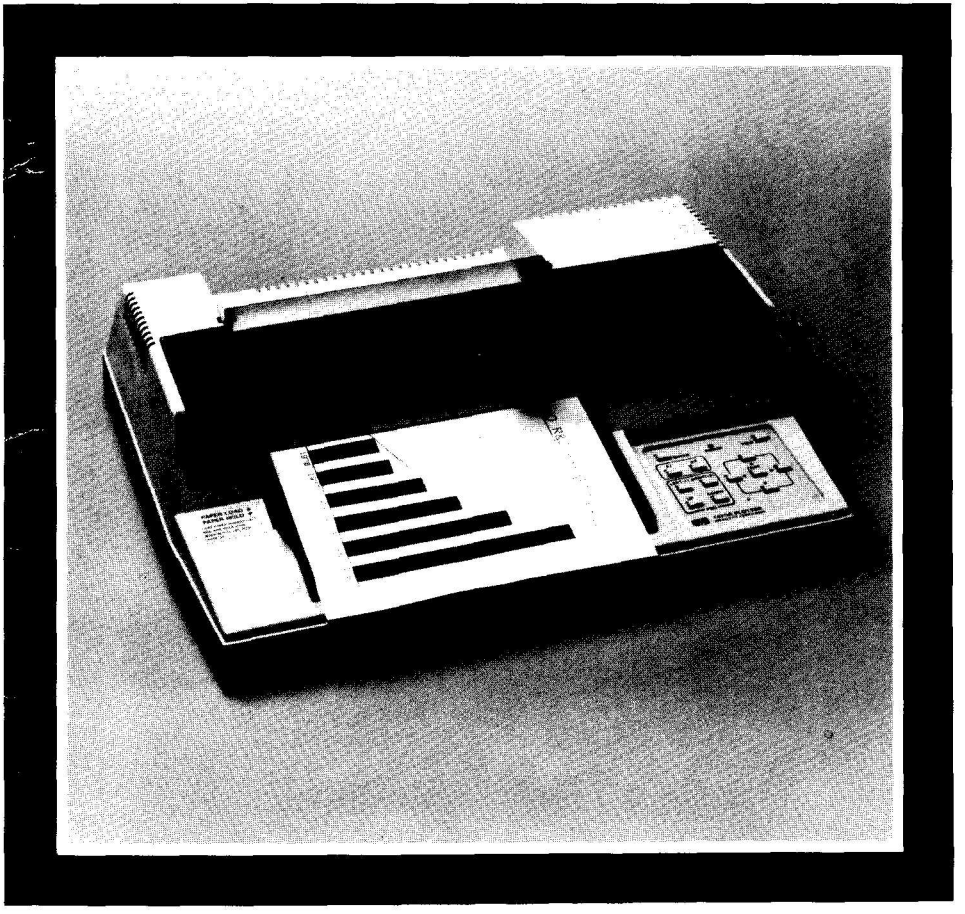
INTERFACING AND PROGRAMMING MANUAL

HP 7470A
Graphics Plotter

RS-232-C/CCITT V.24

DESIGNED FOR HP-IB SYSTEMS

HP-IL



The United States Federal Communications Commission (in 47 CFR 15.838) has specified that the following notice be brought to the attention of users of this product.

FEDERAL COMMUNICATIONS COMMISSION RADIO FREQUENCY INTERFERENCE STATEMENT

"This equipment generates and uses radio frequency energy and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio and television reception. It has been type tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

— reorient the receiving antenna
— relocate the computer with respect to the receiver
— move the computer away from the receiver
— plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet prepared by the Federal Communications Commission helpful:

'How to Identify and Resolve Radio-TV Interference Problems'.

This booklet is available from the US Government Printing Office, Washington, DC 20402, Stock No. 004-000-00345-4."

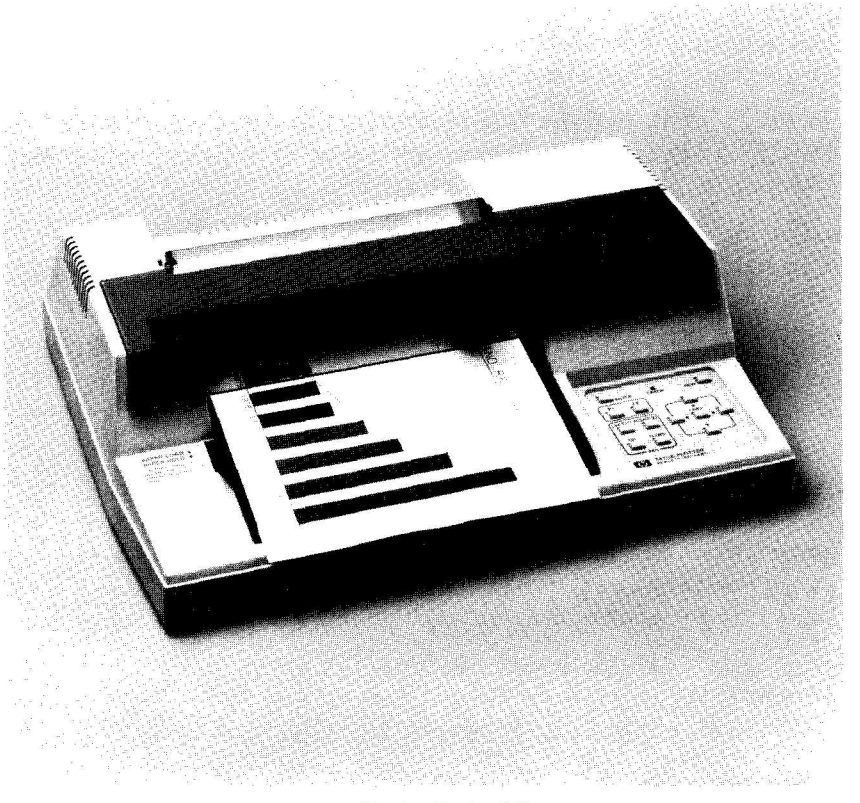
INTERFACING AND PROGRAMMING MANUAL

HP 7470A
Graphics Plotter

RS-232-C/CCITT V.24

DESIGNED FOR HP-IB SYSTEMS

HP-IL



©1982, 1984, by Hewlett-Packard Company
16399 W. Bernardo Drive, San Diego, CA 92127-1899

Manual Summary

Chapter 1: Getting Started

Contains information concerning manual usage, a description of the plotter, its interfaces, the HP-GL language, and three instructions.

Chapter 2: Establishing Boundaries and Units

Explains the concept of plotting area, plotter and user units, scaling, and the instructions used to set and output the scaling points and window, and to scale the plotting area.

Chapter 3: Controlling the Pen and Plotting

Describes the instructions for pen control and vector plotting.

Chapter 4: Enhancing the Plot

Describes instructions for drawing tick marks and differentiating traces.

Chapter 5: Labeling

Describes the instructions used in labeling to set direction, size, and slant of characters, as well as instructions for character set and label terminator selection and for designing your own characters.

Chapter 6: Digitizing

Describes the instructions used to digitize with the plotter and demonstrates how to check for the presence of a digitized point.

Chapter 7: Obtaining Information from the Plotter

Describes the instructions used to obtain information about pen position, errors, and capabilities of the plotter.

Chapter 8: Putting the Commands to Work

A step-by-step example illustrating the procedures to be followed to draw labels and plot data using HP-GL instructions.

Chapter 9: HP-IB Interfacing

Summarizes operation of the plotter with the Hewlett-Packard Interface Bus (HP-IB) and explains the methods for addressing and sending and receiving data over the interface bus.

Chapter 10: RS-232-C/CCITT V.24 Interfacing

Describes how to connect the plotter with a terminal and/or computer, summarizes the methods for establishing a handshake protocol between the plotter and computer, and explains the device control instructions that are used to set up and control the handshake protocol.

Chapter 11: HP-IL Interfacing

Describes the Hewlett-Packard Interface Loop (HP-IL) and explains the methods for sending and receiving data over the interface loop.

Appendix A: An HP-IB Overview

Provides an overview of the Hewlett-Packard Interface Bus (HP-IB).

Manual Summary (Continued)

Appendix B: Instruction Syntax

Provides a summary of both HP-GL and device control instructions.

Appendix C: Reference Material

Includes a summary of default conditions, error messages, scaling equations, NOP instructions, ASCII codes, and character sets.

Table of Contents

Chapter 1: Getting Started 1-1
What You'll Learn in This Chapter 1-1
HP-GL Instructions Covered 1-1
Terms You Should Understand 1-1
How to Use HP 7470 Documentation 1-2
For First Encounters with the 7470 1-3
For First Encounters with HP-GL 1-3
For Experienced HP-GL Programmers 1-3
Understanding Manual Conventions and Syntax 1-3
A Brief Look at the 7470 Plotter 1-5
The 7470 Plotter's Instruction Set 1-6
HP-GL Syntax 1-6
How to Use the Examples in This Manual 1-10
The Default Instruction, DF 1-10
The Initialize Instruction, IN 1-11
The Input Mask Instruction, IM 1-12
Looking Ahead 1-15

Chapter 2: Establishing Boundaries and Units 2-1
What You'll Learn in This Chapter 2-1
HP-GL Instructions Covered 2-1
Terms You Should Understand 2-1
The Plotting Area 2-2
Unit Systems 2-3
The Plotter Unit 2-3
User Units 2-3
Setting the Scaling Points 2-3
Setting P1 and P2 Manually 2-4
The Input P1 and P2 Instruction, IP 2-4
The Output P1 and P2 Instruction, OP 2-5
The Scale Instruction, SC 2-6
The Input Window Instruction, IW 2-9
The Output Window Instruction, OW 2-10
Advanced Programming Tips 2-11

Chapter 3: Controlling the Pen and Plotting 3-1
What You'll Learn in This Chapter 3-1
HP-GL Instructions Covered 3-1
Terms You Should Understand 3-1

Table of Contents (Continued)

Chapter 3: Controlling the Pen and Plotting (Continued)
The Pen Instructions, PU and PD 3-2
The Select Pen Instruction, SP 3-2
The Velocity Select Instruction, VS 3-3
The Plot Absolute Instruction, PA 3-4
The Plot Relative Instruction, PR 3-8
Plotting with Variables 3-11
The Circle Instruction, CI 3-12
The Arc Absolute Instruction, AA 3-17
The Arc Relative Instruction, AR 3-19

Chapter 4: Enhancing the Plot 4-1
What You'll Learn in This Chapter 4-1
HP-GL Instructions Covered 4-1
The Tick Instructions, XT and YT 4-2
The Tick Length Instruction, TL 4-2
The Symbol Mode Instruction, SM 4-4
The Line Type Instruction, LT 4-6

Chapter 5: Labeling 5-1
What You'll Learn in This Chapter 5-1
HP-GL Instructions Covered 5-1
Terms You Should Understand 5-1
Plotter Character Sets 5-2
The Designate Standard Character Set Instruction, CS 5-3
The Designate Alternate Character Set Instruction, CA 5-4
The Select Standard Set Instruction, SS 5-4
The Select Alternate Set Instruction, SA 5-5
The Define Terminator Instruction, DT 5-6
The Label Instruction, LB 5-7
Labeling with Variables 5-9
The Absolute Direction Instruction, DI 5-10
The Relative Direction Instruction, DR 5-11
Spacing Between Characters 5-12
The Character Plot Instruction, CP 5-13
The Absolute Character Size Instruction, SI 5-15
The Relative Character Size Instruction, SR 5-16
The Character Slant Instruction, SL 5-18

Table of Contents (Continued)

Chapter 5: Labeling (Continued)
The User Defined Character Instruction, UC 5-19
Parameter Interaction in Labeling Commands 5-21
Advanced Programming Tips 5-27

Chapter 6: Digitizing 6-1
What You'll Learn in This Chapter 6-1
HP-GL Instructions Covered 6-1
Terms You Should Understand 6-1
Preparing Your Plotter for Use as a Digitizer 6-2
The Digitize Point Instruction, DP 6-2
The Digitize Clear Instruction, DC 6-3
The Output Digitized Point and Pen Status Instruction, OD 6-3
Digitizing with the 7470 6-4
Manual Method 6-4
Monitoring the Status Byte 6-5
HP-IB Interrupts and Polling 6-7

Chapter 7: Obtaining Information From the Plotter 7-1
What You'll Learn in This Chapter 7-1
HP-GL Instructions Covered 7-1
Terms You Should Understand 7-1
A Brief Word about Plotter Output 7-2
Notes for an HP-IB User 7-2
Notes for an RS-232-C User 7-2
Notes for an HP-IL User 7-2
The Output Actual Position and Pen Status Instruction, OA 7-3
The Output Commanded Position and Pen Status Instruction, OC 7-4
The Output Error Instruction, OE 7-5
The Output Factors Instruction, OF 7-6
The Output Identification Instruction, OI 7-7
The Output Options Instruction, OO 7-7
The Output Status Instruction, OS 7-8
Summary of Output Response Types 7-9

Table of Contents (Continued)

Chapter 8: Putting the Commands to Work 8-1
What You'll Learn in This Chapter 8-1
Problem 8-2
Solution 8-2
Setup and Scaling 8-2
The Axes and Their Labels 8-3
Adding Color and Emphasis 8-5
Plotting Your Data 8-6
Listing 8-9
Advanced Programming Tips 8-10
Filling and Hatching 8-10
Filling a Bar 8-10
Hatching a Bar 8-12
Filling Segments of Pie Charts 8-13

Chapter 9: HP-IB Interfacing 9-1
What You'll Learn in This Chapter 9-1
HP-IB Implementation on the 7470 9-2
Interface Switches and Controls 9-2
Addressing the Plotter 9-2
Bus Commands 9-4
Reaction to Bus Commands DCL, SDC, and IFC 9-4
Serial and Parallel Polling 9-4
Addressing the 7470 as a Talker or Listener 9-6
Computers with No High Level I/O Statements 9-6
Computer with High Level I/O Statements 9-6
Sending and Receiving Data 9-7
Computer-to-Plotter 9-7
Plotter-to-Computer 9-10

Chapter 10: RS-232-C/CCITT V.24 Interfacing 10-1
What You'll Learn in This Chapter 10-1
Setting Up Your RS-232-C Plotter: a Checklist 10-2
Plotter Environments 10-2
Using a Plotter Directly Connected to a Computer Mainframe or Personal Computer 10-2
Using a Plotter in an Environment with a Terminal 10-4
Using the Plotter in a Terminal-only Environment 10-9
Connecting the RS-232-C Interface 10-10

Table of Contents (Continued)

Chapter 10: RS-232-C/CCITT V.24 Interfacing (Continued)
Output Baud Rate 10-12
Stop Bits 10-13
Transmission Errors 10-13
Handshaking 10-14
Software Checking 10-17
Xon-Xoff Handshake 10-19
Enquire/Acknowledge Handshake 10-20
Hardwire Handshake 10-22
RS-232-C Device Control Instructions 10-22
Command Syntax for Device Control Instructions 10-23
The Plotter On Instruction, ESC . (or ESC . Y 10-24
The Plotter Off Instruction, ESC .) or ESC . Z 10-24
The Set Plotter Configuration Instruction, ESC . @ 10-25
The Output Buffer Space Instruction, ESC . B 10-26
The Output Extended Error Instruction, ESC . E 10-27
The Set Handshake Mode 1 Instruction, ESC . H 10-28
The Set Handshake Mode 2 Instruction, ESC . I 10-29
The Abort Device Control Instruction, ESC . J 10-31
The Abort Graphic Instruction, ESC . K 10-32
The Output Buffer Size Instruction, ESC . L 10-32
The Set Output Mode Instruction, ESC . M 10-33
The Set Extended Output and Handshake Mode Instruction, ESC . N 10-34
The Output Extended Status Instruction, ESC . O 10-38
The Reset Handshake Instruction, ESC . R 10-40

Chapter 11: HP-IL Interfacing 11-1
What You'll Learn in this Chapter 11-1
An Overview of HP-IL 11-1
HP-IL Implementation on the 7470 11-2
Reaction to Interface Commands and Messages 11-3
Addressing the Plotter 11-3
Sending and Receiving Data 11-4
Computer-to-Plotter 11-4
Plotter-to-Computer 11-5

Table of Contents (Continued)

Appendix A: An HP-IB Overview A-1
HP-IB System Terms A-1
Interface Bus Concepts A-1
Message Concepts A-2
The HP Interface Bus A-4
HP-IB Lines and Operations A-4
Interface Functions A-7
Bus Messages A-8

Appendix B: Instruction Syntax B-1
HP-GL Syntax B-1
RS-232-C Instruction Syntax B-12

Appendix C: Reference Material C-1
Binary Coding and Conversions C-1
Binary-Decimal Conversions C-1
Scaling Without Using the SC Instruction C-2
Plotter Default Conditions C-5
HP-GL Error Messages C-6
RS-232-C Error Messages C-6
The No Operation Instructions, NOP C-7
ASCII Character Codes C-7

Subject Index SI-1

Notes

Chapter 1
Getting Started

What You'll Learn in This Chapter

In this chapter you will learn what is covered in this manual and what other manuals you may need or find useful. In addition, this chapter contains a description of the plotter and its three available interfaces. The plotter's language and its syntax are described. A table is given showing all the HP-GL instructions implemented in the 7470. At the end of the chapter, three instructions from the plotter's language, HP-GL (Hewlett-Packard Graphics Language) are described.

HP-GL Instructions Covered

DF The Default Instruction
IN The Initialize Instruction
IM The Input Mask Instruction

Terms You Should Understand

HP-GL — Hewlett-Packard Graphics Language — the two-letter-mnemonic graphics language understood by the 7470 Plotter and other HP graphics devices. The instruction's mnemonic is suggestive of its role. For instance, PA is used to plot to absolute coordinates, SP is used to select a pen, and DR is used to establish the relative direction of labeling.

HP-IB — Hewlett-Packard Interface Bus — HP's implementation of IEEE standard 488-1978 digital interface for programmable instrumentation, commonly found on HP desktop computers, and some larger computers. The HP-IB interface is standard on the Option 002 plotter.

RS-232-C/CCITT V.24 Interface — another popular standardized interface. It is commonly found on large computers, personal computers, and in environments where communication over telephone lines is required. The RS-232-C/CCITT V.24 interface is standard on the Option 001 plotter.

HP-IL — Hewlett-Packard Interface Loop — an interface used on some Hewlett-Packard personal computing products to communicate with peripheral devices such as the 7470 plotter. The HP-IL interface is standard on the Option 003 plotter.

How to Use HP 7470 Documentation

This manual contains interfacing and programming information for the HP 7470 Plotter and all its interfacing options. The Option 001 plotter is equipped with the RS-232-C/CCITT V.24 Interface. The Option 002 plotter is interfaced through the Hewlett-Packard Interface Bus (HP-IB) which conforms to ANSI/IEEE 488-1978 specifications. The Option 003 plotter is equipped with the Hewlett-Packard Interface Loop for personal computing devices. All interfaces use the Hewlett-Packard Graphics Language (HP-GL) for control of plotter graphics capabilities. Unless specifically noted, all information in this manual pertains to all configurations.

NOTE: All information in this manual for Option 001 plotters applies equally to RS-232-C and CCITT V.24 interfaces. For purposes of simplicity, both are referred to as RS-232-C. ■

Documentation for this plotter is designed to enable you to use the plotter easily without reading unnecessary manuals. All plotters are shipped with this manual, an Operator's Manual (Part No. 07470-90002), an Interconnection Guide (07470-90003), and a Reference Card (07470-90004). The Operator's Manual contains all information you will need to operate, but not program, the plotter. The Interconnect Guide explains how to physically connect your plotter to certain computers or calculators, and contains instructions for verifying that the connection has been made. The Reference Card contains a list of the plotter's HP-GL instructions with their parameters, its device control instructions for the RS-232-C version, and a list of error numbers and their meanings.

For First Encounters with the 7470

If you have just received your HP 7470, read the Operator's Manual and the Interconnection Guide before attempting to operate the plotter. After inspecting your plotter, its power cord, and accessories as described in the Operator's Manual, refer to the appropriate chapter of this manual for initial setup and addressing or handshaking protocol for your configuration. RS-232-C users should read Chapter 10, HP-IB users should read Chapter 9, and HP-IL users should read Chapter 11.

For First Encounters with HP-GL

If you have never programmed in HP-GL, after reading the interfacing chapter, read Chapters 1 through 5 in order. These chapters describe the instructions you will use in almost every application. Running the examples given with the instructions will help you learn. Next, read Chapter 8 to see how all the instructions work together in a program. When you have an application requiring digitizing or plotter output, read Chapters 6 and 7.

For Experienced HP-GL Programmers

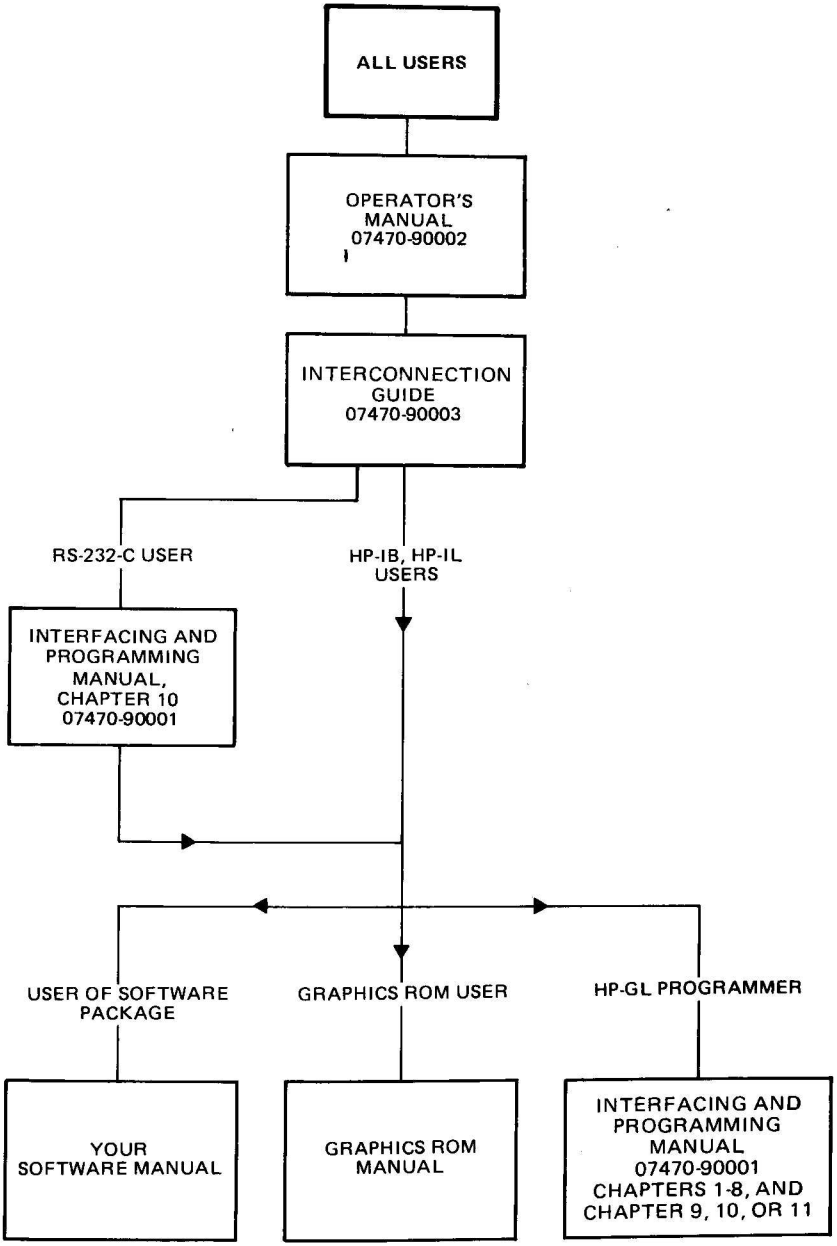
If you are an experienced HP-GL programmer, you may find Appendix B of this manual or the Reference Card most helpful. Since there are differences in syntax between this and other plotters, you should read Chapter 1 of this manual before programming. The 7470 has added capabilities not found in earlier plotters. Among these are the ability to plot to non-integer user-unit values, to mirror labels using negative size and direction parameters, and to output the current window values. To understand these differences, you need to read the sections on scaling (SC, Chapter 2), plotting (PA and PR, Chapter 3), and setting label size and direction in Chapter 5. In the instruction set summary in Appendix B, page numbers for the complete description are listed with each instruction.

Understanding Manual Conventions and Syntax

Before reading any part of this manual, you should understand the meaning of type styles, symbols, and number representation used in text. A detailed explanation of syntax symbols is given in the section entitled HP-GL Syntax in this chapter and Command Syntax for Device Control Instructions in Chapter 10. The following conventions also apply. Words typed in small boldface type are either buttons, switches, or words actually found on the plotter or computer. Headings in **REVERSE** type are used to help locate specific parts of the writeup of an instruction. **REVERSE** type in a smaller size is used to denote 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 instead of commas, counting both to the left and right of the decimal point (54 321.123 45).

Follow the documentation road map below:



A Brief Look at the 7470 Plotter

The HP 7470 Graphics Plotter is a vector plotter which produces high quality, multicolor graphics plots on two sizes of drawing media: English ANSI A (8½ × 11 in.) or metric ISO A4 (210 × 297 mm). With programmable pen velocity and a choice of standard fiber tip or transparency pens, the 7470 can produce distinctive graphics not only on standard paper, but also on other media such as transparency film. The plotter offers both fast plotting speed and high line quality, achieved using Hewlett-Packard's micro-grip drive technology. This technology provides low-inertia grit-covered wheels to move the paper in one axis while the pen moves along the other axis. Plotting occurs with approximately 2 g acceleration and a maximum velocity of 38.1 cm/s (15 in./s). The result is exceptional line and character quality and high throughput. The 7470 has addressable resolution of 0.025 mm (0.001 in.) and repeatability of 0.10 mm (0.004 in.) for any given pen.

The multicolor graphics capability is provided by programmed or front-panel selection of two pens. If additional colors are desired, the program can be paused to allow manual installation of additional pens. Seven different dashed-line fonts and symbol mode plotting provide additional trace identification capabilities.

Character plotting speed of up to six characters per second enables you to produce fully-lettered graphs quickly. Annotation can be easily done using any of five character sets, including three European sets. Text can be written in any direction, with or without character slant, and in varying sizes.

The 7470 is engineered to be especially useful in the areas of business graphics, statistics, medicine, numerical control, surveying, and engineering design. An optional overhead transparency kit enables you to produce high quality graphic transparencies from your plotting programs. For faster comprehension, you can present economic trends, engineering or scientific data, marketing plans, profit data, or sales forecasts pictorially. And with a choice of media, you can create paper hardcopy for an individual's attention or transparencies for group presentations.

Whether data are tabulated, measured, or computed, depend on the reliable 7470 to prepare multicolored plots of excellent line quality and high resolution.

The 7470 Plotter's Instruction Set

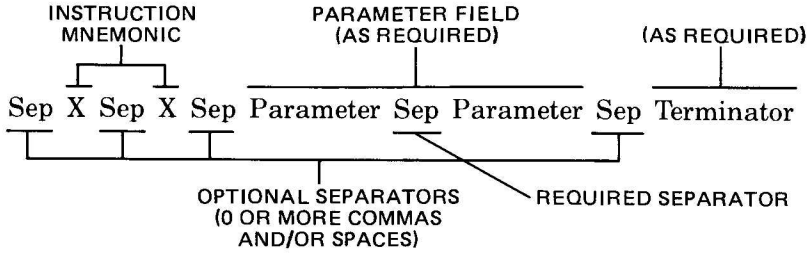
All three interface configurations for the HP 7470 Plotter use the same Hewlett-Packard Graphics Language (HP-GL) instruction set, with minor exceptions.* HP-GL consists of two-letter mnemonic instructions which activate the plotter. A table listing the instructions alphabetically is located at the end of the next section. Syntax descriptions and explanations of these instructions are contained in Chapters 1 through 8. Six additional HP-GL instructions cause no operation but are included for compatibility with other HP plotters. These instructions are listed in Appendix C.

Fourteen additional instructions, called device control instructions, are required by the RS-232-C configuration. These instructions are used to establish plotter output and handshake protocol, and to control conditions which are pertinent only to the RS-232-C environment. In an RS-232-C plotter, all HP-GL instructions enter the plotter's internal buffer and are executed in a first-in, first-out sequence. Device control instructions do not enter the buffer, but instead are executed immediately upon receipt. Refer to Chapter 10 for the syntax description and an explanation of the device control instructions.

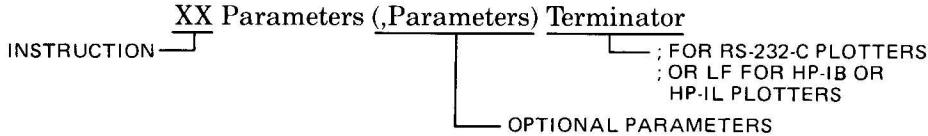
HP-GL Syntax

An HP-GL instruction is a two-letter mnemonic, which may be upper- or lowercase. A command is defined as an instruction followed by its parameter field, if any, and a terminator. If parameters follow the mnemonic, they must be separated from each other by at least one comma or space, or by a + or − sign which may be preceded by commas or spaces. Optional commas and/or spaces may be used as separators before, after, and between the mnemonic and before the terminator. An instruction is terminated by a semicolon, nonalphabetic and nonnumeric characters such as # or $, or by the next mnemonic. If you have an HP-IB or HP-IL plotter, a line feed can also terminate an instruction. (Note that if you have an RS-232-C plotter, a line feed is **not** a valid terminator.) Some instructions will execute immediately after the mnemonic or last required parameter is received. When this is the case, the designation for the terminator is shown in parentheses in the syntax description. The syntax is shown on the next page.

*Option 001 provides 45 instructions; Option 002 provides 42 instructions; Option 003 provides 41 instructions. Refer to the Plotter Instruction Set table in this chapter.



NOTE: The syntax implemented on the 7470 is extremely flexible and differs from that used on other Hewlett-Packard plotters such as the HP 9872. Therefore, any software written for the 7470 which takes advantage of its less rigorous syntax will not be able to drive most other HP plotters. If software is to be used with other HP-GL plotters, the more rigorous syntax of the HP 9872 plotter should be used.



The 9872 syntax does not allow separators between the characters of the mnemonic. One comma must separate parameters. Only ; or LF may be used as the terminator for HP-IB or HP-IL plotters, and only ; may be used as the terminator for RS-232-C plotters. In addition, parameters requiring integer format may not contain a decimal point or decimal fraction. ■

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 exception is the pen parameters in the HP-GL instruction, UC.

The label instruction, LB, is a special case; it must be terminated with the label terminator character. This character defaults to the ASCII end-of-text character, ETX, whose decimal equivalent is 3. The label terminator may be changed from its default value using the define terminator instruction, DT.

The parameter fields must be specified in the format defined by the syntax of each respective HP-GL instruction. The format can be of three types:

1. Integer Format — a parameter in integer format between −32 768.0000 and +32 767.9999. Decimal fractions of parameters which must be integers are truncated. If no sign is specified, the parameter is assumed to be positive.

2. Decimal Format — a number between −128.0000 and 127.9999 with an optional decimal point and decimal fraction with up to four significant digits. If no sign is specified, the parameter is assumed to be positive.

3. Label Fields — any combination of text, numeric expressions, or string variables. Refer to The Label Instruction, LB, Chapter 5, for a complete description.

Some instructions such as PA, PR, PU, and PD may have multiple parameters. Separators are required between these parameters. These optional parameters are shown in parentheses in the syntax descriptions.

The syntax shown under the description of 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.
necessary parameter	All typeset items are required parameters.
( )	All items in parentheses are optional.
c....c	Any number of labeling characters.
(,..)	Any number of X,Y coordinate pairs.
terminator	; or any nonnumeric or nonalphabetic character such as $ or #, or the next mnemonic. LF is also valid for HP-IB and HP-IL plotters.
(terminator)	Terminator for an instruction which will execute after the last necessary parameter is received.

The following table shows the 7470's HP-GL instruction set.

Plotter Instruction Set

Instruction		Description
AA	X,Y, arc angle (, chord angle)	Arc absolute*
AR	X,Y, arc angle (, chord angle)	Arc relative*
CA	n	Designate alternate set n
CI	radius (, chord angle)	Circle*
CP	spaces, lines	Character plot
CS	m	Designate standard set m
DC		Digitize clear
DF		Set default values

Plotter Instruction Set (Continued)

Instruction		Description
DI	run, rise	Absolute direction
DP		Digitize point
DR	run, rise	Relative direction
DT	c	Define label terminator
IM	e(,s(,p))	Input e, s, and p masks
IN		Initialize
IP	$P1_x$,$P1_y$ (, $P2_x$,$P2_y$)	Input P1 and P2
IW	X_{lo},Y_{lo},X_{hi},Y_{hi}	Input window
LB	c....c	Label ASCII string
LT	t(,l)	Designate line type and length
OA		Output actual position and pen status
OC		Output commanded position and pen status
OD		Output digitized point and pen status
OE		Output error
OF		Output factors
OI		Output identification
OO		Output options
OP		Output P1 and P2
OS		Output status
OW		Output window
PA	x,y(,x,y(....))	Plot absolute
PD	(x,y(,...))	Pen down
PR	x,y(,x,y(,...))	Plot relative
PU	(x,y(,...))	Pen up
SA		Select alternate character set
SC	X_{min},X_{max},Y_{min},Y_{max}	Scale
SI	width, height	Absolute character size
SL	tan θ	Absolute character slant (from vertical)
SM	c	Symbol mode c
SP	n	Select pen
SR	width, height	Relative character size
SS		Select standard character set
TL	tp(,tn)	Tick length
UC	(pen,)x,y,pen(,...)	User defined character**
VS	v	Select velocity v
XT		X-axis tick
YT		Y-axis tick

*Available only with Option 001 plotters that have the serial prefix number **2308A** or higher.

**Not available with Option 003.

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. New programmers are strongly encouraged to enter and run all examples. When the example consists of only a few HP-GL commands, these commands are listed in quotes. No line numbers or BASIC statements are included. The literal string listed should be sent to the plotter; the quotation marks only serve to delimit the string and are included because many computer languages define literal strings by placing them inside quotation marks. Do not send the quotation marks to the plotter.

Longer examples are given as programs or program segments in BASIC. The programs will run only if the plotter has been defined as the system printer. Since the statement to do this is highly system-dependent, it is not included (except in Chapter 8). Unless specific mention is made in the text, the BASIC used is that of the HP-83/85. You may need to make slight changes in the BASIC statements for them to run on your computer. You may also need an I/O ROM to obtain output from the plotter. Check with the nearest HP dealer or HP Sales and Support Office. If you are operating in an RS-232-C environment, you will need to establish handshaking protocol and include the necessary device control statements in your program.

If you are programming in another language, substitute the output or input commands of your language for the BASIC statements PRINT and ENTER. Change FOR...NEXT loops and replacement statements (X = 3.14) to whatever statements are comparable in your language. All characters enclosed in quotes in the program listing must be sent to the computer using output statements; in addition, some variables, which are not included in quotes, may need to be sent.

Refer to Chapter 9 for some examples of complete simple programs to send and receive information between the plotter and specific computers in an HP-IB environment. The Interconnection Guide (07470-90003) has some examples of sending HP-GL commands from specific computers; there are examples using RS-232-C, HP-IB, and HP-IL interfaces in that document.

The Default Instruction, DF

DESCRIPTION The default instruction, DF, sets certain plotter functions to a predefined state.

USES The instruction can be used to return the plotter to a known state while maintaining the same settings of P1 and P2. This assures that unwanted graphics parameters such as character size, slant, or scaling are not inherited from another program but that the positions of P1 and P2 remain unchanged.

SYNTAX *DF* terminator

EXPLANATION No parameters are used; a numeric parameter will cause error 2 and the instruction will not execute.

A DF command sets the following plotter functions to the conditions shown in the following table. P1 and P2 are not changed.

Default Conditions

Function	Conditions
Plotting mode	Absolute (PA)
Relative character direction	Horizontal (DR1,0)
Line type	Solid line
Line pattern length	4% of the distance from P1 to P2
Input window	Mechanical limits of plotter
Relative character size	Width = 0.75% of ($P2_x - P1_x$) Height = 1.5% of ($P2_y - P1_y$)
Symbol mode	Off
Tick length	tp = tn = 0.5% of ($P2_x - P1_x$) for Y-tick and 0.5% of ($P2_y - P1_y$) for X-tick
Standard character set	Set 0
Alternate character set	Set 0
Character set selected	Standard
Character slant	0 degrees
Mask value	223,0,0
Digitize clear	On
Scale	Off
Pen velocity	38.1 cm/s (15 in./s)
Label terminator	ETX (ASCII decimal equivalent 3)
Chord angle*	Set to 5 degrees for AA, AR, and CI

*Applicable only to Option 001 plotters that have the serial prefix number **2308A** or higher.

The Initialize Instruction, IN

DESCRIPTION The initialize instruction, IN, returns the plotter's graphics conditions to the initial power-on state by program control. This instruction has no effect on handshake protocol or the plotter's state (programmed on or programmed off) in an RS-232-C environment.

USES The instruction can be used to return the plotter to a known state at the beginning of a graphics program so unwanted graphics parameters such as character size, slant, and scaling are not inherited from another program. P1 and P2 are set to power-on positions.

SYNTAX *IN* terminator

EXPLANATION No parameters are used; a numeric parameter will cause error 2 and the instruction will not execute.

An IN command is the equivalent of switching the plotter off and then on again (except that conditions set by escape code sequences are not changed in an RS-232-C environment). The initialize command sets the plotter to the same conditions as the default command and sets these additional conditions.

- The pen is raised.
- The scaling points P1 and P2 are set to the points P1 = 250,279 and P2 = 10 250,7479.
- All HP-GL errors are cleared. Bit position 3 of the output status byte is set to true(1) indicating the plotter has been initialized. (This bit is cleared by OS.)
- The setting of the **US/A4** switch (for paper size) is read, thus establishing the limits within which the pen can move (mechanical hard clip limits).

The Input Mask Instruction, IM

DESCRIPTION The input mask instruction, IM, controls the conditions under which HP-GL error status is reported, the conditions that can cause an HP-IB service request message, and the conditions that can cause a positive response to an HP-IB parallel poll.

USES With all three interface configurations (HP-IB, HP-IL, and RS-232-C), this instruction can be used to change the conditions under which HP-GL error status is reported. In an HP-IB system only, the instruction is used to enable the plotter to send a service request message when specified bits of the status byte are set, and/or enable a positive response to a parallel poll under the conditions specified.

SYNTAX *IM* E-mask value (,S-mask value (,P-mask value)) (terminator)
or
IM (terminator)

EXPLANATION In both the RS-232-C and HP-IL configurations, the S- and P-masks are of no use and are ignored if present. The E-mask is used by all three configurations.

The E-mask value specified is the sum of any combination of the bit values shown in the following table. When an HP-GL error occurs, the bit in the E-mask corresponding to the error number as shown below is tested to determine if the error bit (bit 5) of the status byte is to be set and the front panel **ERROR** LED is to be turned on. If a bit is not set, there is no way to ever determine if that error occurred.

E-Mask Bit Value	Bit	Error Number	Meaning
1	0	1	Instruction not recognized
2	1	2	Wrong number of parameters
4	2	3	Bad parameter
8	3	4	Not used
16	4	5	Unknown character set
32	5	6	Position overflow
64	6	7	Not used
128	7	8	Vector or PD received with pinch wheels up

The default E-mask value of 223 (128 + 64 + 16 + 8 + 4 + 2 + 1) will specify that all errors except error 6 will set the error bit in the status byte and turn on the **ERROR** LED whenever they occur. Error 6 will not set the error bit or turn on the **ERROR** LED if it occurs, since it is not included in the E-mask value. Errors 4 and 7 never occur so setting the E-mask to 151 will set the same conditions as the default value 223.

The S-mask value specified is the sum of any of the bit values shown below. It determines when a service request message will be sent. 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 message sent. The status of bit 6 changes as plotter conditions change, and is cleared or set as required.

S-Mask Bit Value	Status Bit Number	Meaning
1	0	Pen down
2	1	P1 or P2 changed
4	2	Digitized point available
8	3	Initialized
16	4	Ready for data; pinch wheels down
32	5	Error
64	6	Not used
128	7	Not used

For example, an S-mask value of 4 specifies that when a digitized point is available, setting bit 2, the service request message will be sent. Setting other bits will not send the service request message.

The P-mask value specifies which of the status-byte conditions will result in a logical 1 response to a parallel poll over the HP-IB interface.

P-Mask Bit Value	Status Bit Number	Meaning
1	0	Pen down
2	1	P1 or P2 changed
4	2	Digitized point available
8	3	Initialized
16	4	Ready for data; pinch wheels down
32	5	Error

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.

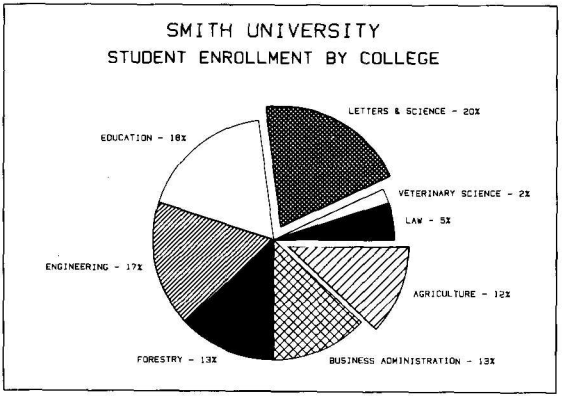
The plotter, when set to default values or initialized, automatically sets the E-mask to 223, the S-mask to 0, and the P-mask to 0. An IM command without parameters or with invalid parameters also sets the masks to the default values 223,0,0.

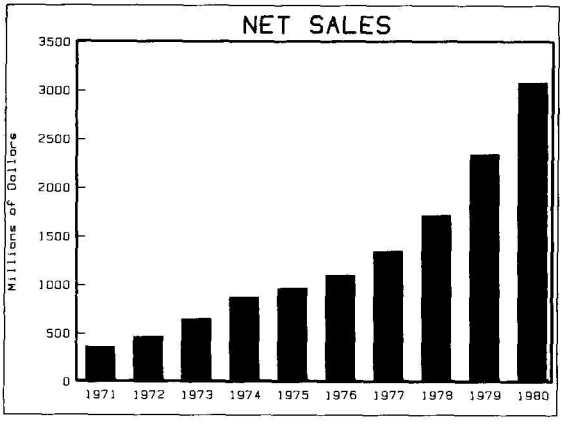
Looking Ahead

Of course you want to use your plotter to create high quality graphic plots. Most plots fall into one of three broad classes: line graphs, bar graphs, or pie charts. Chapter 8 contains a discussion of a line graph. Shown below are a bar graph and a pie chart.

Pie charts are an effective way to show parts of a whole entity; the slices of the pie are the component parts. The pie chart here has some segments "exploded" for emphasis. To construct a pie chart, the data is computed as a percentage of the total and each data value is converted to the appropriate segment of a full 360-degree circle. A simple circle-drawing program is found under the PA instruction in Chapter 3. To create a pie chart you'll need to draw segments of a circle (arcs) and connect the endpoints of the arcs to the circle's center with plotted lines.

There are three types of bar graphs; simple bar graphs, stacked bar graphs, and clustered bar graphs. The simple bar graph here shows that sales are increasing. Bar graphs are essentially a collection of rectangles; i.e., four plotted lines. Each of these rectangles is filled; refer to the advanced programming tips at the end of Chapter 8 to learn how to create a filled or hatched area. A stacked bar might be used to show these same sales data broken down into sales by region. Portions of each bar would be colored or shaded differently to show the sales in each region. Another way of showing sales by region would be to use a separate bar for each region and to "cluster" all the bars for one year together with a larger space between each cluster of bars. There is one cluster for each year of data.





Notes

Chapter 2
Establishing Boundaries and Units

What You'll Learn in This Chapter

In this chapter you will learn about the plotting area, how to define a point in this area, and the two kinds of units used to describe the plotting area. After reading this chapter, you will be able to decide which units to use for your data. In addition, you will be able to scale the plotting area into user units appropriate for your data, and to set or read the current scaling points. You will be able to restrict plotting to only a portion of the plotting area, and read the current limits of the plotting area.

HP-GL Instructions Covered

IP The Input P1 and P2 Instruction
OP The Output P1 and P2 Instruction
SC The Scale Instruction
IW The Input Window Instruction
OW The Output Window Instruction

Terms You Should Understand

Scaling — dividing the plotting area into units convenient for your application. Units need not be the same physical size in both axes, nor do there need to be an equal number of units in the X- and Y-axes.

Scaling Points — the points on the plotting surface moved to when the front panel buttons **P1** and **P2** are pressed. These points are assigned the user-unit values specified by the parameters of the scaling instruction SC.

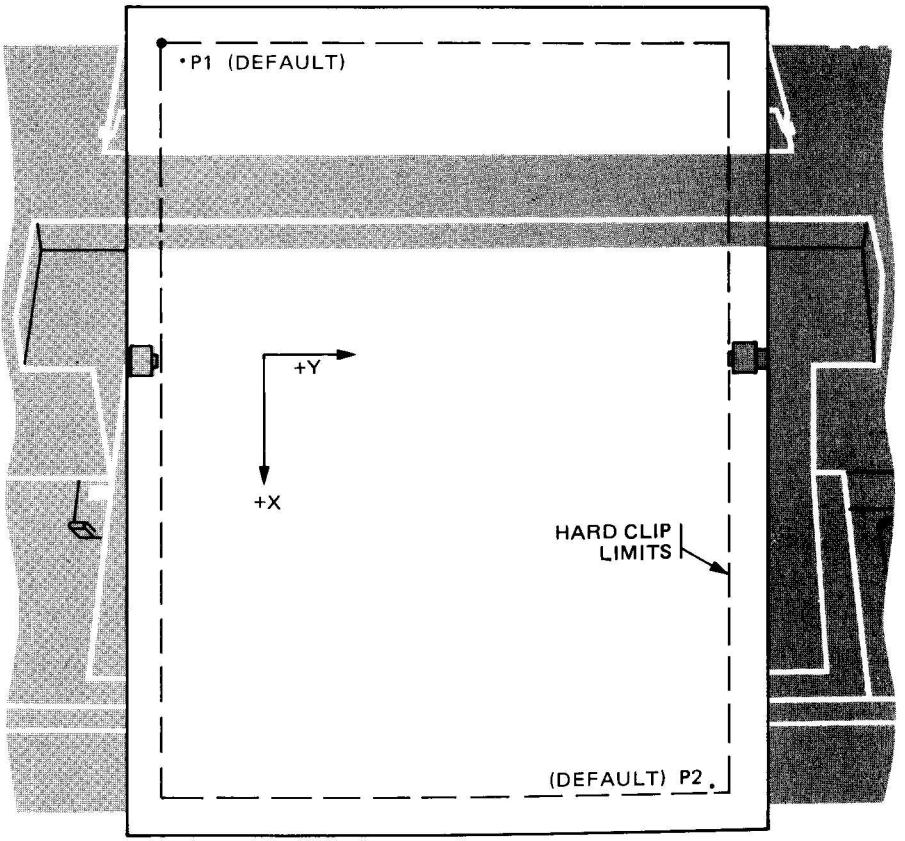
Window — that part of the plotting area in which plotting of points, lines, and labels can occur. At power on, the window is set to the mechanical limits of the plotter. Nothing can be drawn outside the current window.

Clipping — restricting plotting to a portion of the plotting area by establishing a window of a certain size.

The Plotting Area

The plotting area is that area of the paper in which the pen can draw. The maximum plotting area for the HP 7470 Plotter is 191 × 272 mm (7.5 × 10.7 in.) when the paper switch is set to **A4**, and is 191 × 257 mm (7.5 × 10.2 in.) when the paper switch is set to **US**. These plotting areas permit plotting on either metric A4 size paper or English 8½-by-11-inch paper and allow for a margin beween the plotting area and the edges of the paper.

The plotting area should be thought of as a two-dimensional Cartesian coordinate system. Remember, in a two-dimensional Cartesian coordinate system, a point is defined by its X- and Y-coordinates; for example, 200,300 represents a distinct point where X=200 and Y=300. When paper is loaded, the orientation of the X- and Y-axes is established as shown in the following diagram. When looking at the plotter from the front, the origin is located near the upper-left corner of the paper. From now on, we will refer to that corner as lower left, since when a plot is viewed, the minimum point is generally at the lower-left corner of the plot.



Unit Systems

There are two unit systems which can be used to define points in the plotting area: plotter units and user units. Plotter units are always the same size. The size of a user unit depends on the parameters of the SC instruction and the settings of the scaling points, P1 and P2.

The Plotter Unit

The plotting area is divided into plotter units; one plotter unit equals 0.025 mm. There are approximately 40 plotter units per millimetre, or approximately 1000 plotter units per inch. One plotter unit is the smallest move the plotter can make. When the paper switch is set to **A4**, the plotting area contains 10 900 plotter units in X and 7650 plotter units in Y. When the paper switch is set to **US**, the plotting area contains 10 300 plotter units in X and 7650 in Y. While the pen can only plot in the area mentioned above, parameters of plot commands between −32 768 and 32 767 plotter units are understood by the plotter. When plotting in plotter units, only integer values are used; parameters are truncated to integers. Refer to The Plot Absolute Instruction, PA, in Chapter 3.

At power on, upon front-panel reset, and whenever an IN command is sent to the plotter, the scaling point P1 is set to 250,279 plotter units and the scaling point P2 is set to 10 250,7479 plotter units. These settings are independent of the setting of the paper switch.

User Units

The plotting area can also be scaled into user units. This is done with the scale instruction, SC, which assigns values to the scaling points P1 and P2. A user unit may be almost any size. The parameters of the SC instruction are truncated to integers between −32 768 and 32 767. Parameters of plot commands must also be in that range but may be decimal numbers with fractional parts. Decimal fractions are not truncated; as a matter of fact, you can set the scaling points at 0,0 and 1,1 and all your data can be decimal fractions between 0 and 1. You can also use the plot relative instruction to plot to a point which, in user units, is beyond the range ±32 768 as long as its location, expressed as plotter units, is in range. Refer to the plot instructions PA and PR in Chapter 3. You will probably use the SC instruction and user units for most plots.

Setting the Scaling Points

Scaling points P1 and P2 can be set programmatically using the input P1 and P2 instruction, IP, as described in a following section. P1 and P2 can be set manually using front panel controls **ENTER**, **P1**, and **P2**.

Setting P1 and P2 Manually

P2 moves when P1 is moved manually. If you want P2 to be at a specific location, set P1 first and then P2. If you want to establish an area of a certain size onto which the parameters of a scale instruction will be mapped, you may set P2 in the desired location relative to the current P1, and then move P1. P2 will move to a corresponding location so that both the X- and Y-distances between P1 and P2 remain constant. If such a move means the new location of P2 will be beyond the plotting area, either or both coordinates of P2 are set to the plotting limits. In this case, the size of the rectangle established by P1 and P2 will, of course, not remain the same. A detailed description, including illustrations, is contained in the HP 7470 Operator's Manual.

To set P1 or P2 manually:

1. Move the pen to the desired location using the front panel arrow buttons.
2. Press **ENTER** simultaneously with **P1** or **P2**. If **ENTER** is not held down, the pen will merely move to P1 or P2 and no change in the location of P1 or P2 will occur.
3. Check the new locations of the scaling points by pressing **P1**; then press **P2**.

The Input P1 and P2 Instruction, IP

DESCRIPTION The input P1 and P2 instruction, IP, provides the means to relocate P1 and P2 through program control.

USES The IP instruction is often used to ensure that a plot is always the same size, especially when the user and programmer are not the same person. It establishes program control of plot size and label direction. This instruction can also be used to move the scaling points P1 and P2 from their default or current locations; to give mirror images of vectors and labels; to change the size of a user unit, thus reducing or enlarging an image; to change the size or direction of labels when relative character size or direction is in effect; and to set P1 and P2 back to their default locations.

SYNTAX *IP* $P1_x,P1_y$ (, $P2_x,P2_y$) (terminator)
or
IP (terminator)

EXPLANATION The new coordinates of P1 and P2 are specified in the order shown above and must be in absolute plotter units. Parameters should be ≥ 0 and within the maximum plotting area. This means $0 \leq X \leq 10\,300$ when the paper switch is set to **US**; $0 \leq X \leq 10\,900$ if the paper switch is set to **A4**; and $0 \leq Y \leq 7650$ for either setting.

Negative parameters greater than or equal to −32 768 will be set to zero. Parameters outside the maximum plotting area (determined by the setting of the paper switch) but less than 32 767 will be set to the limits of the plotting area. Parameters less than −32 768 or greater than 32 767 will cause error 3 and the coordinates of P1 and P2 will not change.

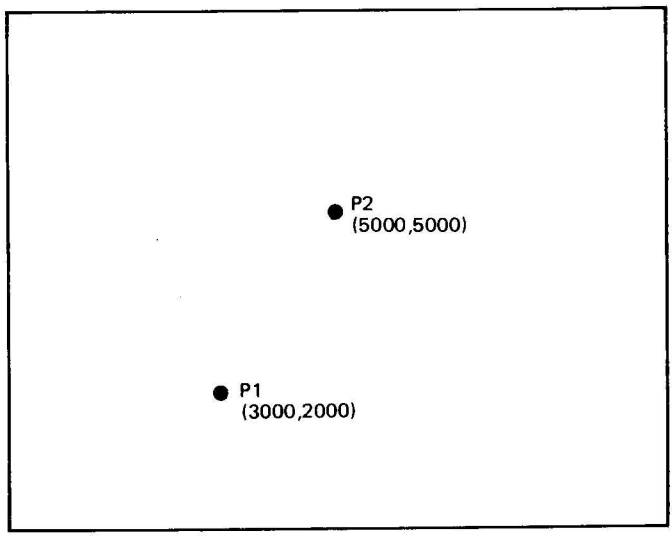
An IP command without parameters will default P1 and P2 to the values 250, 279, 10 250, 7479 regardless of the paper switch setting.

Upon receipt of a valid IP command, bit position 1 of the output status word is set true (1).

Upon power on, front-panel reset, or execution of an IN or DF command, the character size is set relative (SR) to the locations of P1 and P2. Unless an SI command has been entered as part of the program, the character size will be directly affected by the IP command.

The following HP-GL command relocates the scaling points P1 and P2 to the positions shown in the figure.

```
"IP 3000,2000,5000,5000;"
```



The Output P1 and P2 Instruction, OP

DESCRIPTION The output P1 and P2 instruction, OP, provides the means to make the current coordinates of P1 and P2 available for output.

USES The instruction can be used to determine the position of P1 and P2 in plotter units. This information can be used with the input window command, IW, to set the window to P1 and P2 under program control, to compute the number of plotter units per user unit when scaling is on, or to determine the numeric coordinates of P1 and P2 when they have been set manually.

SYNTAX *OP* (terminator)

EXPLANATION After an OP command is received, the plotter will output the coordinates of P1 and P2 in plotter units as four integers in ASCII in the following form:

$$P1_x,P1_y,P2_x,P2_y \quad [TERM]$$

where [TERM] is the output terminator for your system. See Terms You Should Understand in Chapter 7.

The range of the integers is determined by the setting of the paper switch as shown below:

US	A4
$0 \leqslant X \leqslant 10\,300$	$0 \leqslant X \leqslant 10\,900$
$0 \leqslant Y \leqslant 7650$	$0 \leqslant Y \leqslant 7650$

Upon completion of output, bit position 1 of the output status byte is cleared.

The Scale Instruction, SC

DESCRIPTION The scale instruction, SC, establishes a user-unit coordinate system by mapping values onto the scaling points P1 and P2.

USES This instruction is used to enable you to plot in user units convenient to your application. For instance, if your X values represent months, then $X_{min} = 1$ and $X_{max} = 12$. If the values for Y-coordinates all lay between 0 and 10, you might use 0 as Y_{min} and 10 as Y_{max}. By adjusting your minimum and maximum values, you can provide additional room for labeling. If your plot is a 12-month bar chart with Y-coordinates 0 to 10, you might scale the X-axis 0 to 14 so the first and last bars are not at the edge of the graph, and scale the Y-axis 0 to 12 leaving room for a title at the top.

SYNTAX *SC* $X_{min},X_{max},Y_{min},Y_{max}$ (terminator)
or
SC (terminator)

EXPLANATION Executing an SC command without parameters (SC;) turns scaling off and subsequent parameters of plot commands are interpreted as plotter units.

When parameters are used, all four parameters are required. Decimal parameters in an SC command are truncated to integers. The parameters X_{min} and Y_{min} define the user-unit coordinates of P1, and the parameters X_{max} and Y_{max} define the user-unit coordinates of P2. P1 and P2 may be any two opposite corners of a rectangle. Scaling points P1 and P2 retain the assigned user-unit coordinate values until scaling is turned off or another SC command redefines their user-unit coordinate values. Therefore, the physical size of a user unit will change when any change is made in the relative position and distance between P1 and P2.

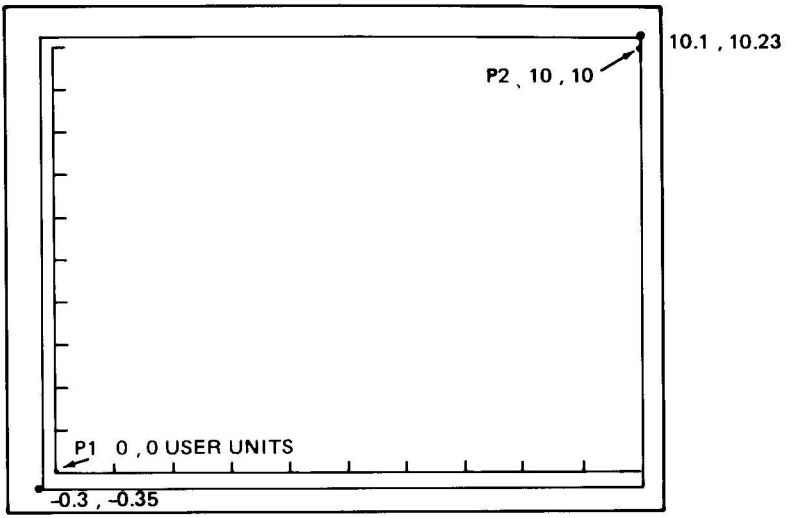
Specifying $X_{max} = X_{min}$ or $Y_{max} = Y_{min}$ or parameters less than −32 768 or greater than 32 767 will turn scaling off. An SC command must have four or no parameters. Otherwise, error 2 will be generated. An SC command which generates an error is ignored and the scaling does not change.

The user-unit coordinate system that is mapped onto the plotter unit coordinate system by the SC command is not limited to the rectangle defined by P1 and P2; it extends over the entire plotting area. When user-unit scaling has been established by executing an SC command with parameters, decimal parameters of plot commands are not truncated; the point 3.5,7.5 is distinct from the point 3.6,7.8. This is different from some other HP plotters and makes plotting of noninteger data much simpler.

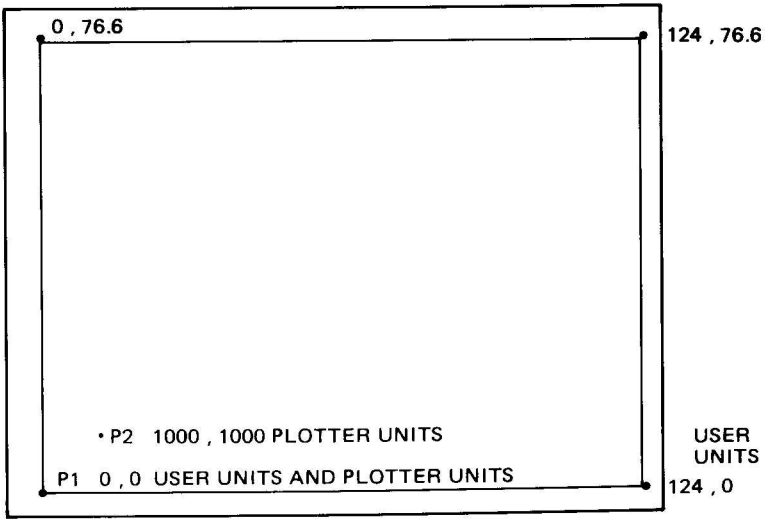
It is not possible to scale an area such that P1 or P2 are assigned values larger than 32 767 or less than −32 768. One way to plot data with values beyond these limits is to reduce your data to acceptable ranges by an arithmetic process before sending it to the plotter. Dividing the data by some factor of 10 so that the integer portions fall between ±32 767 and sending decimal plot parameters is probably the easiest solution.

The illustrations which follow show the coordinate grids mapped onto the plotting area as a result of executing the indicated commands when the paper switch is set to **US**. In all cases, the points labeled at each corner are just outside of the plotting area. If a PA command with these parameters is sent to a plotter with the indicated scaling and the paper switch set to **US**, the pen will move to the corner and lift, indicating the point is outside the plotting area.

"IP; SC 0,10,0,10;"



"IP 0,0,1000,1000; SC 0,10,0,10;"



The Input Window Instruction, IW

DESCRIPTION The input window instruction, IW, provides the means to restrict programmed pen motion to a rectangular area of the plotting surface. This area is called the "window."

USES The instruction can be used to establish a hard clip area, i.e., restrict plotting to a certain area of the paper. The instruction is especially useful when your data should fall in a certain range but your scaling is larger (perhaps you have left room for labels) and you don't want lines outside the normal data area. It is also useful when hatching (shading) rectangular areas.

SYNTAX *IW* $X_{\text{lower left}}$,$Y_{\text{lower left}}$,$X_{\text{upper right}}$,$Y_{\text{upper right}}$ (terminator)
or
IW (terminator)

EXPLANATION Parameters are always interpreted as plotter units. When four parameters are included, the hard clip limits are set according to the parameters. If no parameters are included, the hard clip limits are set to the maximum plotting area. That area was determined by the setting of the rear-panel paper switch as read when the plotter was last initialized by either power up, front-panel reset, or execution of an IN command.

The four parameters specify, in absolute plotter units, the X- and Y-coordinates of the lower-left and upper-right corners of the window area. The parameters should be positive and less than or equal to 10 900 or 10 300 for X (depending on the setting of the paper switch) and less than 7650 for Y. Parameters between −32 768 and 0 are set to 0, and parameters larger than the limits of the absolute plotting area but less than 32 767 are set to 10 300 or 10 900 for X and 7650 for Y. If $X_{\text{lower left}}$ is greater than $X_{\text{upper right}}$ or $Y_{\text{lower left}}$ is greater than $Y_{\text{upper right}}$, no error is set but no plotting can occur.

At power on, or when an IN or DF command is executed, the window is automatically set to the current mechanical limits i.e., maximum plotting area. The window set by DF may not correspond with the current setting of the paper switch if the setting has been changed since power on, a front-panel reset, or the last IN command was executed.

The Output Window Instruction, OW

DESCRIPTION The output window instruction, OW, provides the means to obtain the X- and Y-coordinates of the lower-left and upper-right corners of the area in which plotting can currently occur.

USES The instruction can be used to determine the area in which any plotting will occur. When executed immediately after power on or the execution of a DF or IN command, the command can be used to determine under program control whether the paper switch is set to **US** or **A4**.

SYNTAX *OW* (terminator)

EXPLANATION No parameters are used. Output is in plotter units.

After an OW command is received, the plotter will output the coordinates of opposite corners of the plotting area in plotter units as four integers in ASCII in the following form:

$$X_{\text{lower left}},\ Y_{\text{lower left}},\ X_{\text{upper right}},\ Y_{\text{upper right}}\ \text{[TERM]}$$

where [TERM] is the output terminator for your system. See Terms You Should Understand in Chapter 7.

The range of the integers is determined by the setting of the paper switch as shown below:

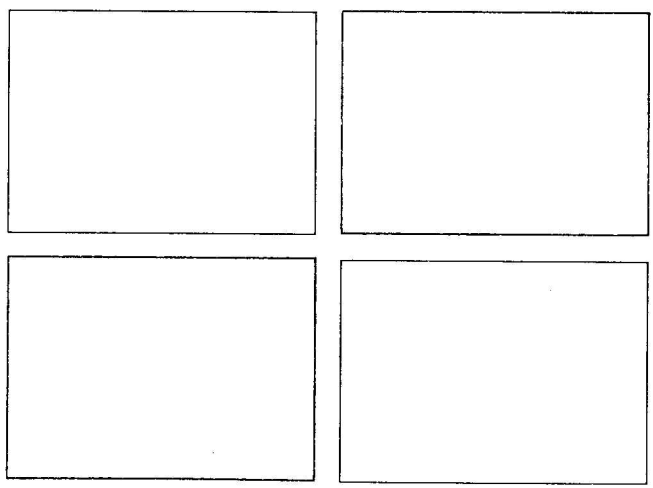
US	**A4**
$0 \leqslant X \leqslant 10\,300$	$0 \leqslant X \leqslant 10\,900$
$0 \leqslant Y \leqslant 7650$	$0 \leqslant Y \leqslant 7650$

If $X_{\text{lower left}}$ is greater than $X_{\text{upper right}}$ or $Y_{\text{lower left}}$ is greater than $Y_{\text{upper right}}$, no window exists in which plotting can occur.

Many software packages read P1 and P2 and use these points to define the maximum plotting area. You may want to obtain the largest plot possible on the 7470. This is the area of the default window, as determined by the setting of the paper switch, not the area established by the default settings of P1 and P2. The first three lines of the following listing will read the window size and set P1 and P2 to these points, so that the largest area possible is used for plotting. In order to change the plotting area, this HP-GL routine should precede the PLOTTER IS statement when programming on HP desktop computers in AGL.

Sometimes you want more than one plot on a page. The rest of the instructions set the window to, and outline four separate areas. A small space has been left between each area by adding or subtracting a constant value from X- and Y-coordinates in the center of the total area. This program could be modified to divide the plotting area into thirds or into areas of any other size. Another application of windowing is shading rectangular areas for bar graphs. See Advanced Programming Tips, Chapter 8.

```
"IN;OW"
 !INSERT LINE TO READ COORDINATES INTO A,B,C,D
"IP";A,B,C,D
"IW";A,B;C/2-100;D/2-100;"SP1;PA";A;B
"PD";C/2-100;B;C/2-100;D/2-100;A;D/2-100;A;B;"PU"
"IW";C/2+100;B;C;D/2-100;"SP2;PU";C/2+100;B
"PD";C;B;C,D/2-100;C/2+100;D/2-100;C/2+100;B;"PU"
"IW";C/2+100;D/2+100;C;D;"SP1;PA";C/2+100;D/2+100
"PD";C;D/2+100;C;D;C/2+100;D;C/2+100;D/2+100;"PU;"
"IW";A;D/2+100;C/2-100;D;"PU;SP2;PA";A;D/2+100
"PD";C/2-100;D/2+100;C/2-100;D;A;D;A;D/2+100;"SP0;"
```



Reduced Plot

Notes

Chapter 3
Controlling the Pen 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 either of the two pens or change pens, how to set and change pen velocity, how to raise and lower the pen, and how to plot. You will learn how to plot to absolute X,Y coordinates or to plot relative to the last pen position. Finally, you will learn how to send variables as parameters of plot commands; this will enable you to write general purpose graphics programs.

HP-GL Instructions Covered

SP	The Select Pen Instruction	CI*	The Circle Instruction
VS	The Velocity Select Instruction	AA*	The Arc Absolute Instruction
PU/PD	The Pen Up/Down Instructions	AR*	The Arc Relative Instruction
PA	The Plot Absolute Instruction		
PR	The Plot Relative Instruction		

Terms You Should Understand

Absolute Plotting — plotting to a point whose location is specified relative to the origin (0,0). When the PA command is used to plot to a point, the pen always moves to the same point on the plotting surface, no matter where the pen was before the move.

Relative Plotting — plotting to a point whose location is specified relative to the current pen position. The point moved to then becomes the effective origin for the next parameter of a plot relative command. When the PR command is used to plot to a point, the destination of the pen depends on where the pen was when the command was received.

Plotter Unit Equivalent — the X,Y coordinates of a point, given in user units, if they were expressed in plotter units.

*Available only with Option 001 plotters that have the serial prefix number **2308A** or higher.

The Pen Instructions, PU and PD

DESCRIPTION The pen up instruction, PU, and the pen down instruction, PD, raise and lower the pen.

USES The instructions are used to raise and lower the pen during plotting. They may be used with parameters to plot or move to the points specified by the parameters.

SYNTAX

PU (terminator)
or
PD (terminator)
and
PU X,Y(,...)(terminator)
or
PD X,Y(,...)(terminator)

EXPLANATION When no parameters are included, the pen up instruction, PU, raises the pen without moving it to a new location. The pen down instruction, PD, lowers the pen without moving it to a new location, if the pen is within the window. If parameters are included, the pen will move, in order, to the X,Y coordinates specified. The coordinates are interpreted as plotter units if scaling is off and user units if scaling is on. Moves are either relative or absolute, depending on whether a PA or PR was the last plot command executed.

If parameters are included, both coordinates of an X,Y coordinate pair must be given. An odd number of parameters will set an error condition, but all X,Y pairs which precede the unmatched parameter will be plotted. For a description of the PU and PD commands with parameters, refer to The Plot Absolute Instruction, PA, and The Plot Relative Instruction, PR, which follow.

NOTE: The plotter has an automatic pen lift feature which will lift the pen after it has been in the pen-down state for 55 seconds and no pen-down plot commands or label commands have been sent to the plotter or no front-panel pen-down moves have been made for 55 seconds. ■

The Select Pen Instruction, SP

DESCRIPTION The select pen instruction, SP, selects and/or stores one of the two pens.

USES The instruction is used to load a pen into the pen holder so that drawing will occur. It can be used to select a pen of a different color or width, during the plotting program. It can be used with a zero parameter or no parameter to store the pen currently in the pen holder into its stall at the end of a program.

SYNTAX *SP* pen number (terminator)
or
SP (terminator)

EXPLANATION The pen parameter must be in the range of −32 768 to 32 767. Decimal fractions are truncated. An odd-numbered parameter selects the pen from the left stall; an even-numbered parameter selects the pen from the right stall. A zero parameter or no parameter stores the pen. When a pen parameter is less than −32 768 or greater than 32 767, an error is generated and the pen does not change.

The Velocity Select Instruction, VS

DESCRIPTION The velocity select instruction, VS, specifies the pen-down speed for plotting and labeling operations.

USES The instruction is used to set velocity to a speed other than the default velocity of 38.1 cm/s and to change the acceleration from its default value of 2 g (980 cm/s^2). This instruction should be used to slow velocity to 10 cm/s when plotting on transparency film. A slightly thicker line can be created by slowing down the pen speed on any medium. A pen nearing the end of its life will write with a clearer, sharper, more solid line if the velocity is slowed.

SYNTAX *VS* pen velocity (terminator)
or
VS (terminator)

EXPLANATION A VS command without parameters sets pen velocity to its default velocity of 38.1 cm/s (15 in./s) and acceleration to 2 g (980 cm/s^2). A VS command with parameters sets the pen velocity for horizontal or vertical pen-down moves to the value specified by the first parameter and slows the acceleration to 0.5 g. Anything after the first parameter is ignored. Parameters must be in the range 0 to 127.9999. A velocity of 0 is set to 0.38 cm/s. Velocity can be set in increments of 0.38 cm/s. Parameters are rounded to the nearest multiple of 0.38 cm/s. Negative parameters and parameters greater than or equal to 97 set an error condition (error 3) and the velocity does not change. Parameters between 38.1 and 96 set velocity to its default value of 38.1 cm/s.

When either the horizontal or vertical velocity falls in the range 0.38 to 3.8 cm/s, it is reset to a slower or faster velocity to avoid this range. This is done to assure lines of high quality. The change is most noticeable when a line is almost vertical or almost horizontal. Pen-down moves will be at the specified velocity except when such adjustment is necessary.

Execution of a VS command with a parameter of 38.1 will slow the acceleration, giving the highest line quality at that maximum speed.

A default instruction, DF, or an initialize instruction, IN, will also reset the velocity and acceleration to the values 38.1 cm/s and 2 g.

The Plot Absolute Instruction, PA

DESCRIPTION The plot absolute instruction, PA, moves the pen to the point(s) specified by the X- and Y-coordinate parameters.

USES The instruction can be used together with PD to draw lines or with PU to move the pen to a specific point on the plot. The instruction can be executed without parameters to establish absolute plotting, as opposed to relative plotting for PU or PD commands with parameters. In this case, the parameters of PU and PD are interpreted as absolute X,Y coordinates until any PR command is received.

SYNTAX *PA* X_1coordinate,Y_1 coordinate (,X_2 coordinate,Y_2 coordinate,...,X_n coordinate, Y_n coordinate)(terminator)

or

PA (terminator)

EXPLANATION Recommended parameters are decimal numbers between −32 768.0000 and 32 767.9999. When scaling is off, parameters are truncated to integers as follows:

- For positive numbers, the fractional portion is discarded and the integer portion remains unchanged. For example, both 1234.4 and 1234.9 become 1234.
- For negative numbers, the fractional portion is discarded and the integer portion is changed to the next more negative integer. For example, both −1234.4 and −1234.9 become −1235. Since you cannot plot to negative values unless scaling is on, (in which case decimal portions of parameters are used), the only time you will observe this is when you use the output commanded position and pen status instruction, OC, and the last X- and/or Y-parameter sent was negative.

NOTE: If you have an HP-IB or RS-232-C plotter that has the serial prefix number **2308A**, or higher, or if you have an HP-IL plotter, you will not observe this truncation with the OC instruction. In these plotters, the OC instruction returns decimal parameters instead of integer parameters when scaling is in effect. ■

When scaling is on, any fractional portion of a parameter is used.

A PA command without parameters sets absolute plotting mode for PU and PD commands with parameters.

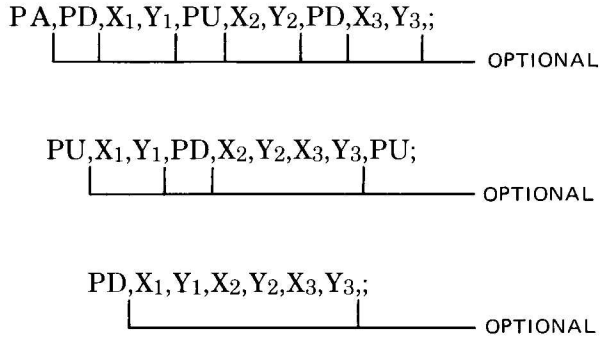
When parameters are included with a PA command, both coordinates of an X,Y coordinate pair must be given. An odd number of parameters

will set an error condition but all X,Y pairs which precede the unmatched parameter will be plotted.

The X-coordinate specifies, in either plotter units or user units, the absolute X-location to which the pen will move. The Y-coordinate specifies, in either plotter units or user units, the absolute Y-location to which the pen will move. If scaling is on, coordinates are in user units. If scaling is off, coordinates are in plotter units.

The mnemonics PU and PD can be included ahead of, between, or after X,Y coordinate pairs. PU lifts the pen; PD lowers the pen.

Any number of coordinate pairs, as well as PU or PD mnemonics, can be listed after a PA instruction. (This is limited only by the ability of the controller to output without a line feed character which is an instruction terminator.) The pen will move to each point in the order given. Commas, spaces, or a sign are required between numeric parameters and are optional after two-letter mnemonics. The last entry is followed by the terminator. In the following examples, commas are used to show optional and required separators. Optional commas or spaces which can be used between each letter of the mnemonics are not shown. The semicolon is used to indicate the terminator.



If no pen control parameter is given, the pen will assume the pen state (up or down) of the previous statement. The PU or PD mnemonics can also be substituted for the PA (or PR) mnemonic. This is equivalent to having PU; or PD; preceding the PA or PR instruction. Therefore, PU and PD with parameters are interpreted to be in place of PA or PR, depending upon which mnemonic, PA or PR, was last specified.

PA is specified by any of the following:

- power-up,
- execution of an IN command,

- execution of a DF command, or
- execution of a PA instruction with or without parameters.

The pen moves and draws lines only within the currently defined window. Refer to The Input Window Instruction, IW, in Chapter 1.

The plotter discards parameters which are out of range. Error 3 will be set (parameter out of range). A PA command with out-of-range parameters will still establish plot absolute mode for future occurrences of PU or PD with parameters. When scaling is off, in-range parameters are greater than or equal to −32 768 and less than or equal to 32 767. When scaling is on, both the parameters and their plotter unit equivalent must also be in that same range. To find the plotter unit equivalent, use the equations in the section Scaling Without Using the SC Instruction in Appendix C.

There are four types of vectors that can be drawn with a PA command from a given last point to some new point.

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

In type two, the pen moves from the last point toward the new point and stops where the line between the two points intersects the current window. The pen up/down condition is as programmed until the intersection is reached. Then, the pen is raised.

In type three, the pen moves with the pen up, to the point where the straight line between the last and new point intersects the window limit. When the pen reaches this point, the pen assumes its programmed (up or down) position. The pen then moves to the new point.

In type four, no pen movement occurs unless the straight line between the last and new point intersects the window. The X- and Y-coordinates of the current pen position are updated. If part of the vector is in the window area, the pen moves, pen up, to the point where the line between the last and the new point first intersects the window limit. The pen moves under programmed pen up/down control to the intersection of the vector and the other window limit. At this point, the pen stops and lifts.

Since out-of-range points are discarded, the plotter will draw a line between the two points on either side of discarded points. You can be sure all lines on your plot represent actual data if you:

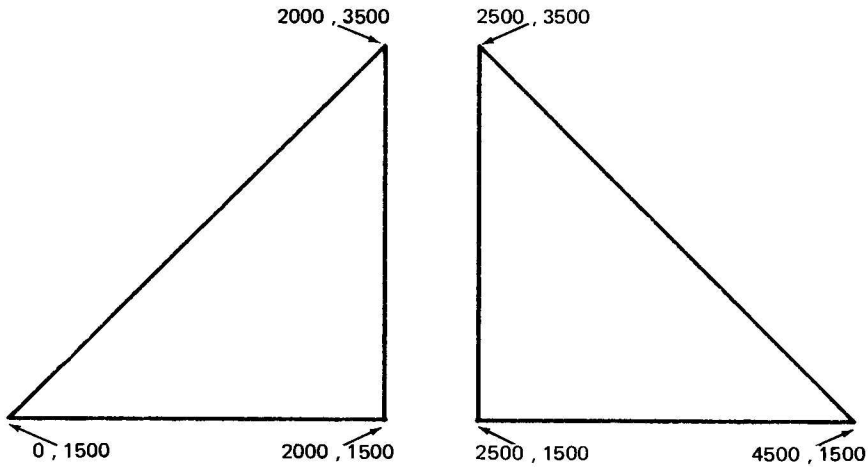
1. have not changed the error mask from its default setting;

2. have not executed an output error instruction; and
3. the error light is not on at the end of your plot.

(The fact that the error light is on does not necessarily mean out-of-range data has been encountered; an error in any HP-GL command will turn the light on.)

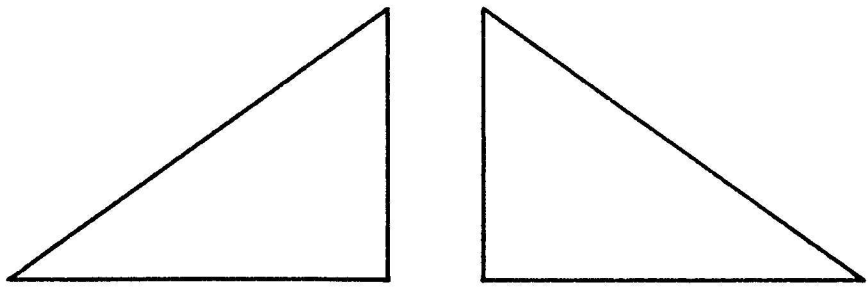
The following strings of HP-GL instructions, if sent to the plotter using a suitable output statement such as PRINT or OUTPUT, will draw two triangles and then move to the point 10 900, 7650 with the pen up.

```
"IN;SP1;"
"PA2000,1500,PD,0,1500,2000,3500,2000,1500,PU,2500,1500;"
"PAPD4500,1500,2500,3500,2500,1500,PU,10900,7650;"
```



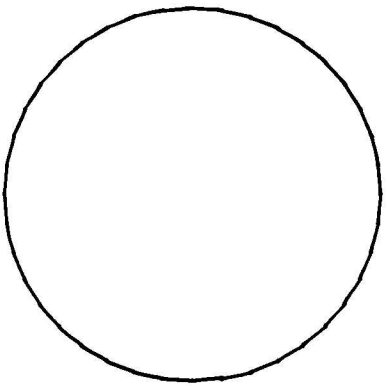
The next strings of HP-GL instructions scale the plotting area into user units 0 to 100 in each axis and again draws two triangles. Use an output statement implemented on your computer to send the strings to the plotter.

```
"IN;SP1;SC0,100,0,100;"
"PA20,15,PD,0,15,20,35,20,15,PU,25,15;"
"PAPD45,15,25,35,25,15,PU;"
```



This final example scales a square plotting area from 0 to 1 in each axis and draws a unit circle. This program should run on most BASIC systems. Change line 10 as necessary for your computer to define the plotter as the system printer. Also, if PI is not a function recognized by your computer, add a line before line 30 to define PI as a variable (PI = 3.1416). Lines 60 and 65 are necessary to limit the number of digits in the X- and Y-coordinates. This prevents the possibility of coordinates being sent to the plotter in scientific notation, which sets an error in the plotter.

```
10 PRINTER IS 705,80
20 PRINT "IN;IP4000,3000,5000,4000;SP1;SC0,1,0,1;"
30 FOR T=0 TO 2*PI+PI/20 STEP PI/20
40 X=COS(T)
50 Y=SIN(T)
60 PRINT USING 65;"PA",X,Y,"PD;"
65 IMAGE 2A,2(MD.DDDD),3A
70 NEXT T
80 PRINT "PU;SP0;"
90 END
```



The Plot Relative Instruction, PR

DESCRIPTION The plot relative instruction, PR, moves the pen relative to its current location by the number of units specified by the X- and Y-increment parameters.

USES The plot relative instruction can be used as PA to draw lines and move to a point. However, with PR, pen movement is relative to the current pen position. The instruction can be executed without parameters to establish relative plotting as opposed to absolute plotting for PU or PD commands with parameters. It is often used to draw multiple occurrences of some figure on a plot, for example, to draw several rectangles of the same size.

SYNTAX *PR* X_1 increment, Y_1 increment(,X_2 increment,Y_2 increment,..., X_n increment, Y_n increment) (terminator)
or
PR (terminator)

EXPLANATION Recommended parameters are in integer format between −32 768.0000 and 32 767.9999. Their plotter unit equivalents should also be in the same range. When scaling is off, parameters are truncated to integers in the manner described under the plot absolute instruction. When scaling is on, any fractional portion of a parameter is used.

A PR command requires that both increments of an X,Y pair be given. An odd number of parameters will set an error condition but all X,Y pairs which precede the unmatched parameter will be plotted.

The X-increment specifies, in either plotter units or user units, the number of units the pen will move in the direction of the X-axis. The Y-increment specifies, in either plotter units or user units, the number of units the pen will move in the direction of the Y-axis. The sign of the parameter determines the direction of movement; a positive value moves the pen in the positive direction and a negative value moves the pen in the negative direction. If scaling is on, both parameters are interpreted as user units. If scaling is off, both parameters are interpreted as plotter units.

The mnemonics PU and PD can be included ahead of, between, or after X,Y increment pairs. PU lifts the pen; PD lowers the pen. Any number of increment pairs, as well as PU or PD mnemonics, (limited only by the ability of the controller to output without a line feed character) can be listed after the PR instruction. The placement of optional or required separators and the terminator is the same as for the PA instruction.

If no pen control parameter is given, the pen will assume the pen state (up or down) of the previous statement. The PU or PD mnemonics can also be substituted for the PR (or PA) mnemonic. This is equivalent to having PU; or PD; preceding the PR or PA command. Since the power-on default is absolute plotting mode, a PR instruction must be executed before parameters of PD or PU commands will be interpreted as X,Y increments. Relative plotting mode is cancelled by execution of a PA, IN, or DF instruction.

The pen moves and draws lines only within the currently defined window. Refer to The Input Window Instruction, IW, Chapter 1. Drawing of vectors in relation to the window is as described under the PA instruction.

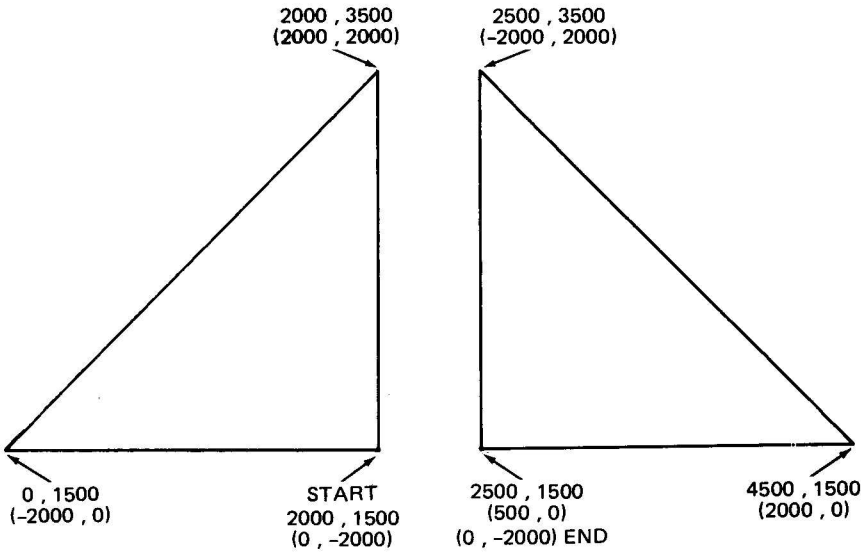
The plotter discards parameters which are out of range or whose plotter unit equivalent would be out of range if the indicated move were made.

Error 3 will be set (parameter out of range). A PR command with out of range parameters will still establish relative plotting mode for future occurrences of PD or PU with parameters.

When scaling is off, in-range parameters are between −32 768 and 32 767. When scaling is on, in-range parameters and their plotter unit equivalent must be between −32 768 and 32 767. To find plotter unit equivalents, refer to the section Converting from User Units to Plotter Units in Appendix C.

The following strings of HP-GL instructions, when sent to the plotter using your computer's output statements, cause triangles to be drawn that are identical to the ones previously drawn using only the PA instruction. The numbers in parentheses on the plot are the X,Y increments of the PR commands. The numbers without parentheses are the plotter unit coordinates of the vertices.

```
"IN;SP1;"
"PA2000,1500,PD,PR-2000,0,2000,2000,0,-2000,PU,500,0;"
"PD2000,0,-2000,2000,0,-2000,PU;"
```



Plotting with Variables

In many plotting applications, it is necessary to plot using variables rather than fixed numbers to define the X- and Y-coordinate values. The values of all HP-GL statement parameters have the same restrictions (integer or decimals in a valid range) when sent as variables as when sent as literals (fixed numbers). The terminators and delimiters of HP-GL statements must be sent to the plotter too. The method of defining output format and variable precision varies from computer to computer. Refer to your computer manual for the appropriate format statements that may be needed in your program.

The following BASIC program illustrates the use of variables in plotting a circle and shows how PRINT statements can be used to send variables as parameters of HP-GL commands. You will use a similar method if you are programming in another language. Quotation marks are used by many computers and languages to delimit literal characters. Note the comma in line 70, which is part of the HP-GL statement to be sent to the plotter; it is specified as a literal in quotes. With the 7470, a space may be substituted for the literal comma, shown in quotes. If your computer automatically sends spaces between variables, these spaces will delimit the coordinate parameters and a literal comma or space will not be necessary. Since scaling is turned on in line 20, the fractional portions of the variables X and Y are used by the plotter. When the plotter is not in scaled mode, fractional portions are truncated by the plotter. Unless you are writing software to be compatible with other HP plotters such as the 9872, it is not necessary to add a formatting statement to assure variables are sent as integers by your computer.

To run this program, be sure to change line 10 as necessary for your computer to define the plotter as the system printer. Also, if PI is not a function recognized by your computer, add a line before line 40 to define PI as a variable (PI = 3.1416).

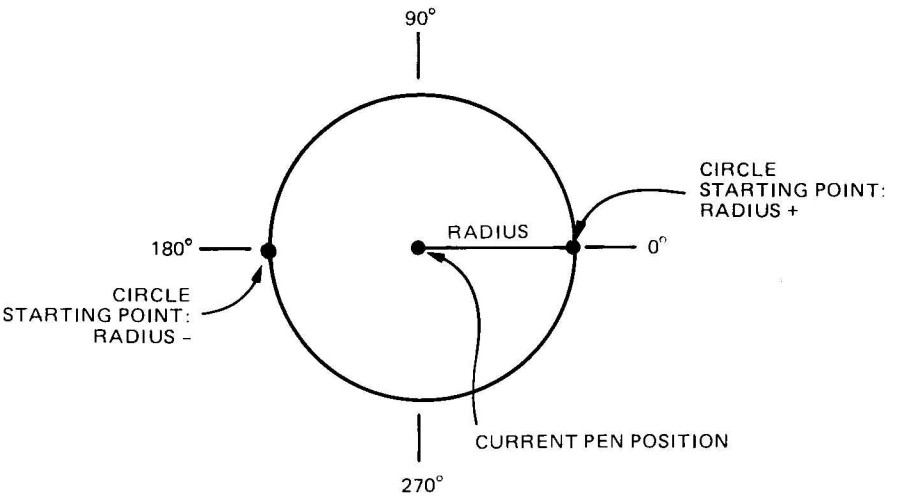
```
10  PRINTER IS 705,80
20  PRINT "IN;SP1;IP1000,1000,6000,6000;"
30  PRINT "SC0,25000,0,25000;"
40  FOR T=0 TO 2*PI+PI/20 STEP PI/20
50  X=4.5*1000*COS(T)+12500
60  Y=4.5*1000*SIN(T)+12500
70  PRINT "PA",X,",",Y,";PD;"
80  NEXT T
90  PRINT "PU;SP0;"
100 END
```

The Circle Instruction, CI

DESCRIPTION The circle instruction, CI, provides the means to draw a circle of a specified radius and chord angle. It is only included in the instruction set of RS-232-C plotters that have the serial prefix number **2308A** or higher.

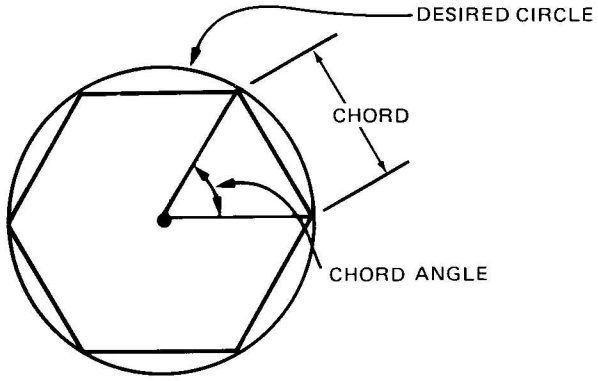
USES The instruction can be used to generate circles with a single command. All computations are internal to the plotter to reduce computer overhead.

SYNTAX *CI* radius (, chord angle) terminator



EXPLANATION The radius parameter can be a positive or negative number in integer format. Its sign 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.

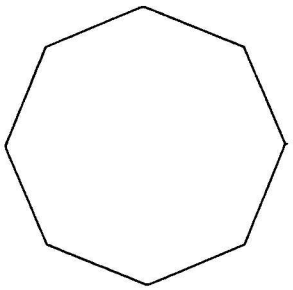
The chord angle parameter is in integer format and governs the smoothness of the circle. It is interpreted as degrees and sets the maximum angle subtended by a chord that is drawn to represent an arc segment of the circle, as shown below. The actual angle used may be changed by the plotter so that all chords are the same length. The sign of the parameter is ignored, except to set the maximum in-range limit to −32 768 or +32 767.



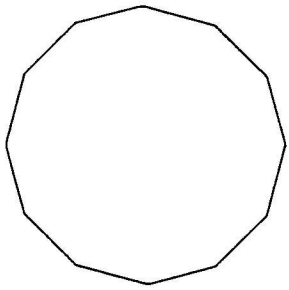
The most useful chord angle values range from 0 to 180; where 0 produces the smoothest circle and larger numbers progressively reduce the number of chords used. Values from 180 to 360 work just the opposite; i.e., larger numbers progressively increase the number of chords used and 360 produces the smoothest circle. This pattern follows modulo 360 through the permitted range of −32 768 to −32 767. Specifying out-of-range parameters sets error 3 and the command is ignored.

The following strings of HP-GL instructions, when sent to the plotter using your computer's output statements, show the effect of different chord angles.

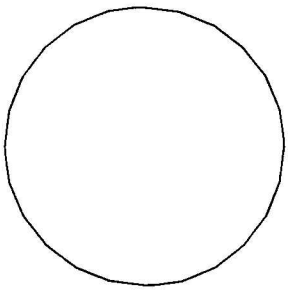
```
"IN;SP1;IP2650,1325,7650,6325;"
"SC-100,100,-100,100;"
"PA-50,40;CI30,45;"
"PA50,40;CI30,30;"
"PA-50,-40;CI30,15;"
"PA50,-40;CI30,5;"
```



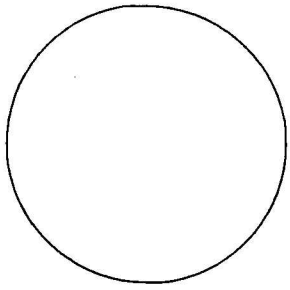
45 DEGREE CHORD ANGLE



30 DEGREE CHORD ANGLE



15 DEGREE CHORD ANGLE



5 DEGREE CHORD ANGLE

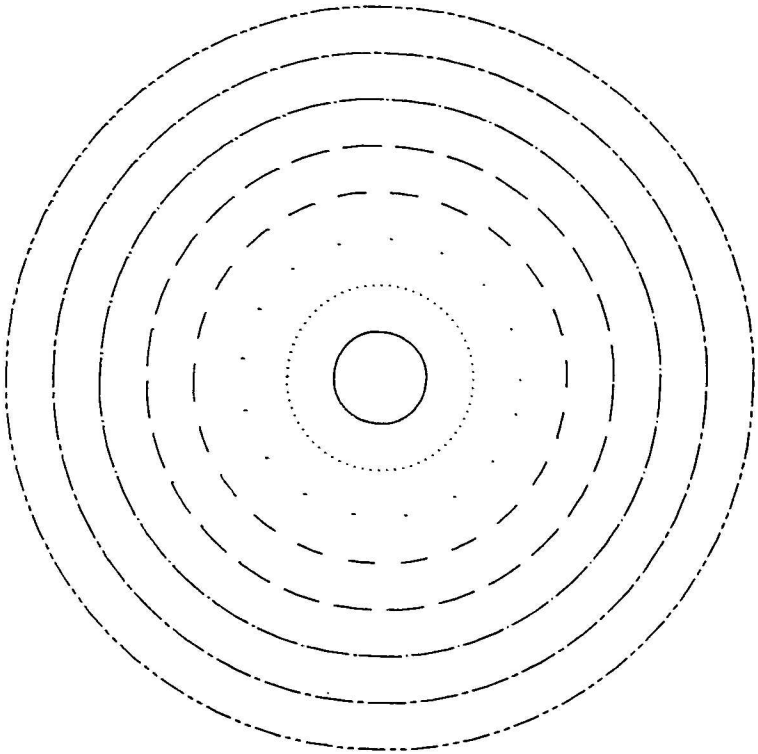
The circle instruction includes an automatic pen down feature. When a circle command is received, the pen lifts (if it was down), moves from the center of the circle to the circle starting point on the circumference, lowers the pen, draws the circle, then returns, pen up, to the center of the circle. After drawing the circle, the pen assumes the pen state (up or down) that was in effect prior to the circle command. To avoid drawing lines to the center of the circle, move to and away from the circle's center with the pen up.

Circles are drawn within the defined window, with clipping occurring outside the window limits. Drawing circles within the window conforms to the definitions given for plotting under the PA instruction.

Each chord of the circle is drawn using the currently defined line type. Refer to The Line Type Instruction, LT, in Chapter 4.

To demonstrate some of the features of the circle instruction, the following strings of HP-GL instructions draw various circles with different line types, radii, and starting points.

```
"IN;SP1;IP2650,1325,7650,6325;"
"SC-100,100,-100,100;"
"PA0,0;LT;CI10,5;LT0;CI-20,5;LT1;CI30,5;"
"LT2;CI-40,5;LT3;CI50,5;LT4;CI-60,5;LT5;
 CI70,5;LT6;CI80,5;"
```



The following BASIC program shows that the circle instruction can also be used to define a series of circles that must be repeated in a particular pattern.

```
10   PRINTER IS 10
20   PRINT "IN;SP1;IP2650,1325,7650,6325;"
30   PRINT "SC-1000,1000,-1000,1000;"
40   PRINT "PA-800,800;"
50   GOSUB 130
60   PRINT "PA200,800;"
70   GOSUB 130
80   PRINT "PA-800,-200;"
90   GOSUB 130
100  PRINT "PA200,-200;"
110  GOSUB 130
120  END
130  PRINT "CI50;PR600,0;CI50;PR-300,-300;CI250;"
140  PRINT "PR-300,-300;CI50;PR600,0;CI50;"
150  RETURN
```

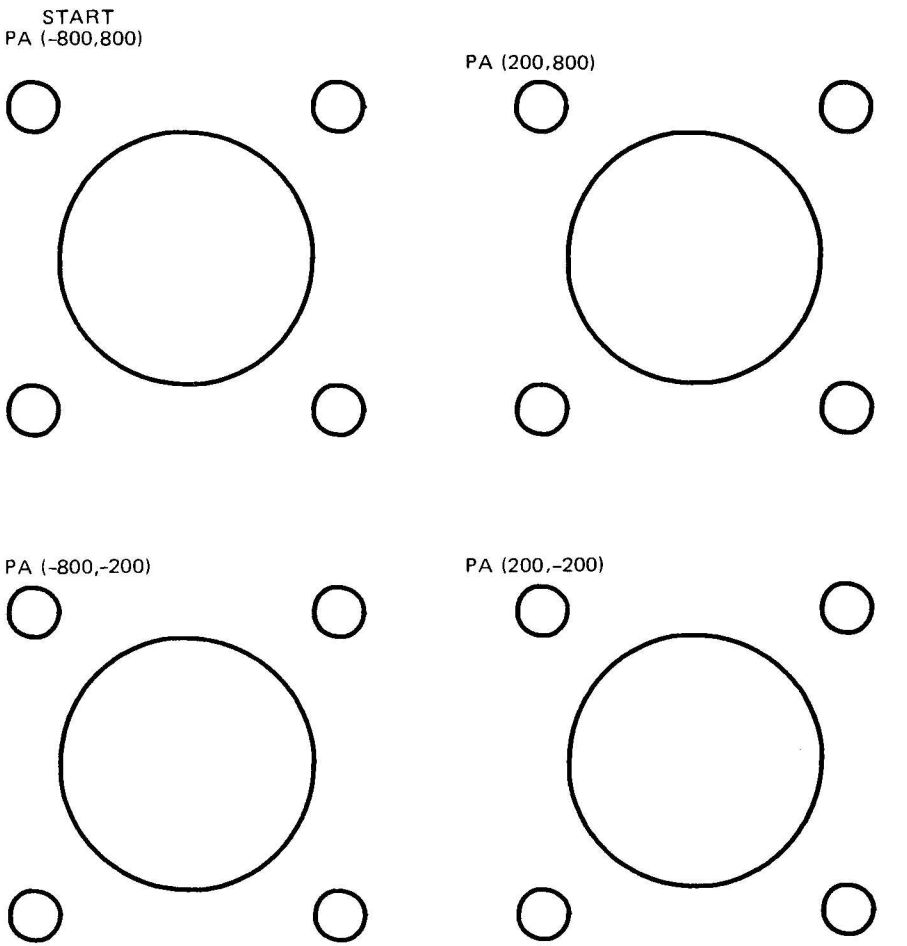
Line 10 defines the select code of the interface; change this statement as necessary for your computer.

Lines 20 and 30 define the plotting area and perform user-unit scaling.

Line 40 moves the pen to point (−800,800) to locate the starting point of the first pattern.

Lines 130 and 140 contain the subroutine necessary to draw the pattern. First, a 50-unit radius circle is drawn, followed by a relative move of 600 units in the X-direction where another 50-unit radius circle is drawn. A move of −300 units in X and −300 units in Y locates the center of the 250-unit circle. The last two 50-unit circles are drawn with the moves shown in the listing.

Lines 60, 80, and 100 locate the starting points of the other three patterns.

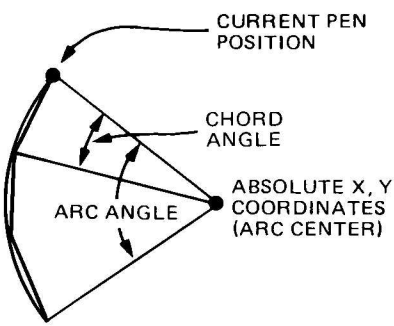


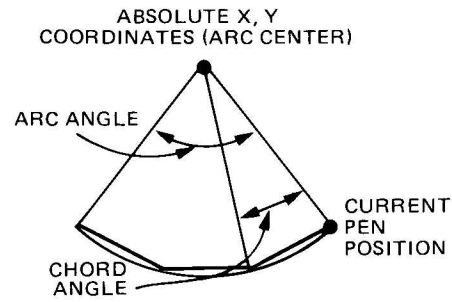
The Arc Absolute Instruction, AA

DESCRIPTION The arc absolute instruction, AA, provides the means to draw an arc with the center point located at a specified absolute point. The arc can be drawn clockwise (CW) or counterclockwise (CCW), subtends the specified arc angle, and conforms to the specified or default chord angle. It is only included in the instruction set of RS-232-C plotters that have the serial prefix number **2308A** or higher.

USES The instruction can be used to draw an arc of any radius, length, and smoothness with a single command. The arc is drawn from the current pen position, and its center point is located by absolute X,Y coordinates.

SYNTAX *AA* X-coordinate, Y-coordinate, arc angle (, chord angle) terminator





EXPLANATION The AA instruction requires that both X- and Y-coordinates be specified (coordinate pair) in integer format. They are interpreted as plotter units if scaling is off or as user units if scaling is on. The X- and Y-coordinates locate the center of the arc and may be located on or off the plotting surface. The current pen position is the starting point of the arc.

The arc angle is in integer format. It is the angle, in degrees, through which the arc is drawn: a positive arc angle draws CCW from the current pen position; a negative arc angle draws CW from the current pen position.

The chord angle parameter is in integer format and governs the smoothness of the arc in the same way as defined under the circle instruction, CI. The sign of the parameter is ignored, except to set the maximum in-range limit to −32 768 or +32 767. The default chord angle is 5 degrees.

Unlike circles, arcs are drawn using the previously commanded pen state (up or down) and line type. If no pen state has been commanded

since initialization, pen up is assumed. If no line type has been commanded, a solid line is drawn.

Arcs are drawn within the defined window, with clipping occurring outside the window limits. Drawing arcs within the window conforms to the definitions given for plotting under the PA instruction.

All parameters must be integers in the range −32 768 to 32 767. Specifying out-of-range parameters sets error 3 and the command is ignored.

The following BASIC program demonstrates the use of the AA instruction.

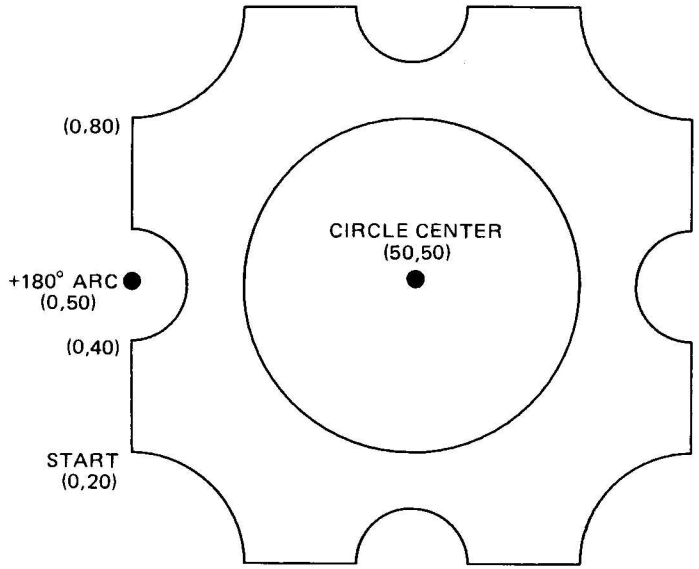
```
10   PRINTER IS 10
20   PRINT "IN;SP1;IP2650,1325,7650,6325;"
30   PRINT "SC0,100,0,100;"
40   PRINT "PA0,20;"
50   PRINT "PD;PA0,40;AA0,50,180;PA0,80;"
60   PRINT "AA0,100,90;PA40,100;AA50,100,180;PA80,100;"
70   PRINT "AA100,100,90;PA100,60;AA100,50,180;PA100,20;"
80   PRINT "AA100,0,90;PA60,0;AA50,0,180;PA20,0;AA0,0,90;"
90   PRINT "PU;PA50,50;CI30;"
100  END
```

Line 10 defines the select code of the interface; change this statement as necessary for your computer.

Lines 20 and 30 initialize the plotter and establish user-unit scaling.

Lines 40 and 50 move the pen to the point 0,20, lower the pen, and draw to the point 0,40, where a 180-degree arc is drawn counterclockwise, centered at 0,50. The pen is then instructed to draw to the point 0,80.

Lines 60 through 90 continue drawing the figure, clockwise, back to the point 0,20, and finish with the circle centered at the point 50,50.

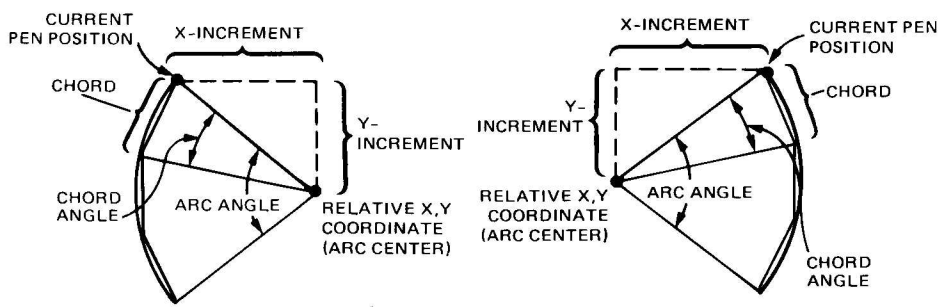


The Arc Relative Instruction, AR

DESCRIPTION The arc relative instruction, AR, provides the means to draw an arc with the center point located relative to the present pen position. The arc can be drawn clockwise (CW) or counterclockwise (CCW), with a specified arc angle and chord angle. It is only included in the instruction set of RS-232-C plotters that have the serial prefix number **2308A** or higher.

USES The instruction can be used to draw an arc of any radius, length, and smoothness with a single command. The arc is drawn from the current pen position, and its center point is located by relative X,Y coordinates.

SYNTAX *AR* X-increment, Y-increment, arc angle (, chord angle) terminator



EXPLANATION The AR instruction requires that both X- and Y-increment parameters (coordinate pair) and arc angle be specified. Increment parameters are in integer format and are interpreted as plotter units if scaling is off or user units if scaling is on. The X- and Y-increment parameters locate the center of the arc with respect to the present pen position. The signs of the increment parameters determine the relative location of the center of the arc. A positive value locates that center in a positive direction and a negative value locates that center in a negative direction. The current pen position is the starting point of the arc.

The arc center can be located on or off the plotting surface. The arc angle is in integer format. It is the angle, in degrees, through which the arc is drawn; a positive arc angle draws CCW; a negative arc angle draws CW.

The chord angle parameter is in integer format and governs the smoothness of the arc in the same way as defined under the circle instruction, CI. The sign of the parameter is ignored, except to set the maximum in-range limit to −32 768 or +32 767. The default chord angle is 5 degrees.

Unlike circles, arcs are drawn using the previously commanded pen state (up or down) and line type. If no pen state has been commanded since initialization, pen up is assumed. If no line type has been commanded, a solid line is drawn.

Arcs are drawn within the defined window, with clipping occurring outside the window limits. Drawing arcs within the window conforms to the definitions given for plotting under the PA instruction.

All parameters must be integers in the range −32 768 to 32 767. Specifying out-of-range parameters sets error 3 and the command is ignored.

The following BASIC programs demonstrate the use of the AR instruction.

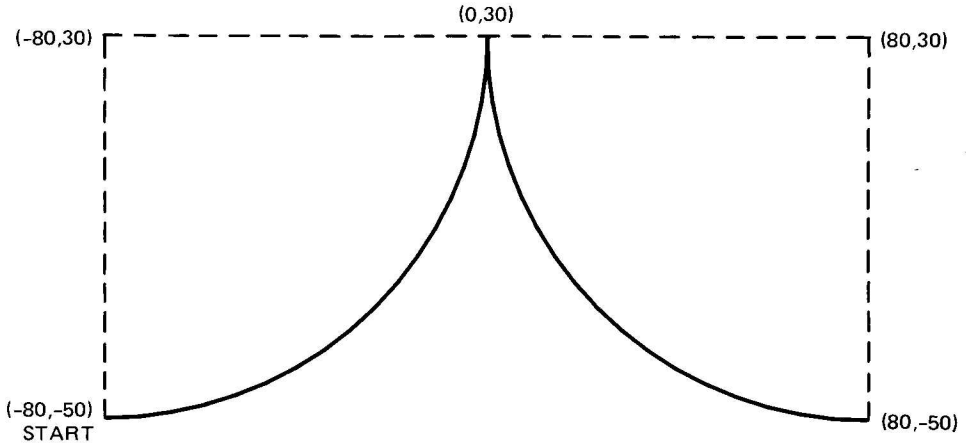
```
10   PRINTER IS 10
20   PRINT "IN;SP1;IP2650,1325,7650,6325;"
30   PRINT "SC-100,100,-100,100;"
40   PRINT "PA-80,-50;PD;AR0,80,90;AR80,0,90;PU;"
50   END
```

Line 10 defines the select code of the interface; change this statement as necessary for your computer.

Line 20 enters the P1 and P2 points on which to scale the plotting area.

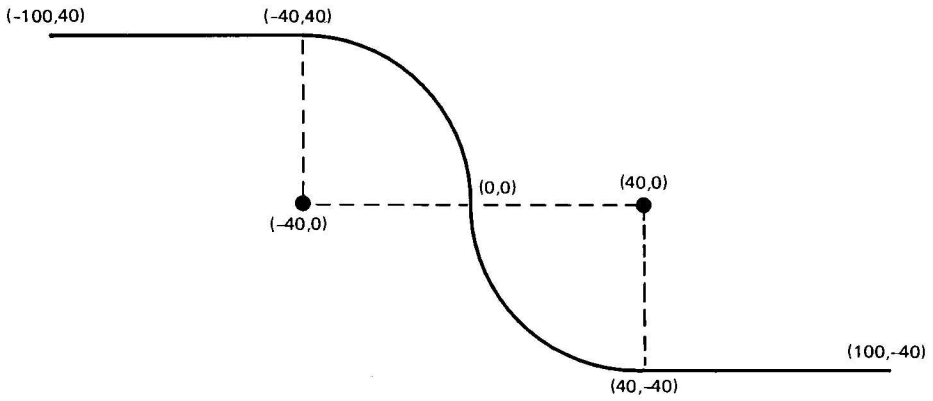
Line 30 scales the plotting area into user units.

Line 40 moves the pen to the point −80,−50, draws a 90-degree CCW arc centered 0,80 units relative to the present pen position, then draws a 90-degree arc centered 80,0 units relative to the 0,30 absolute pen position. Note that a pen down command, PD, is required to draw the arc.




```
10 PRINTER IS 10
20 PRINT "IN;SP1;IP2650,1325,7650,6325;"
30 PRINT "SC-100,100,-100,100;"
40 PRINT "PA-100,40;PD;PR60,0;AR0,-40,-90;AR40,0,90;PR60,0;PU;"
50 END
```

In this example, line 40 moves the pen to the point −100,40, lowers the pen, and plots 60,0 units relative to the previous pen position, −100,40. It then draws a 90-degree CW arc centered at 0,−40 units relative to the new −40,40 pen position, and follows it with a 90-degree CCW arc centered 40,0 units relative to the 0,0 pen position, the endpoint of the first arc. Finally, it plots 60,0 units relative to the pen position 40,−40, the endpoint of the second arc.



Notes

Chapter 4
Enhancing the Plot

What You'll Learn in This Chapter

Now that you can draw lines, you are ready to create your own plots. In this chapter you will learn how to enhance your plots by using HP-GL instructions to draw tick marks on axes or create grids, draw a symbol or character of your choice at each data point, and draw dashed or dotted lines. All these enhancements will make 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

The Tick Instructions, XT and YT

DESCRIPTION The tick instruction, XT, draws a vertical X-tick at the current location. The tick instruction, YT, draws a horizontal Y-tick at the current pen location.

USES These instructions can be used to draw tick marks on axes, draw grid lines by making the tick length 100%, or draw horizontal or vertical lines either centered on or ending at the current pen position.

SYNTAX *XT* (terminator)
or
YT (terminator)

EXPLANATION Neither instruction requires parameters; numeric parameters are ignored. The terminator should be included to complete the command.

The tick mark will be drawn at the current pen position whether the pen is up or down.

The tick length is specified by the tick length instruction, TL. If no tick length is specified, the length defaults to 0.5% of ($P2_x - P1_x$) for YT or 0.5% of ($P2_y - P1_y$) for XT for each (positive and negative) portion of the tick. Refer to The Tick Length Instruction, TL, which follows.

The following example draws a horizontal line 3000 plotter units long, places X-ticks at the endpoints and at X-locations 1200 and 2200, and raises and stores the pen.

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



The Tick Length Instruction, TL

DESCRIPTION The tick length instruction, TL, specifies the length of the tick marks drawn by the plotter. The tick lengths are specified as a percentage of the horizontal and vertical distances between the scaling points P1 and P2.

USES The instruction can be used to set the length of both positive and negative portions of tick marks. The instruction can be used with only one parameter to suppress the negative portion of a tick mark, or with a first parameter of zero to suppress the positive portion of the tick. Setting the tick length, tp, to 100 enables the user to draw grids easily, using XT and YT instructions.

SYNTAX *TL* tp (,tn) (terminator)
or
TL (terminator)

EXPLANATION Both parameters must be between −128 and +127.9999. Use of positive parameters is recommended. For most applications, parameters will be between 0 and 100.

The up and right 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, taking P1 as the lower-left corner.

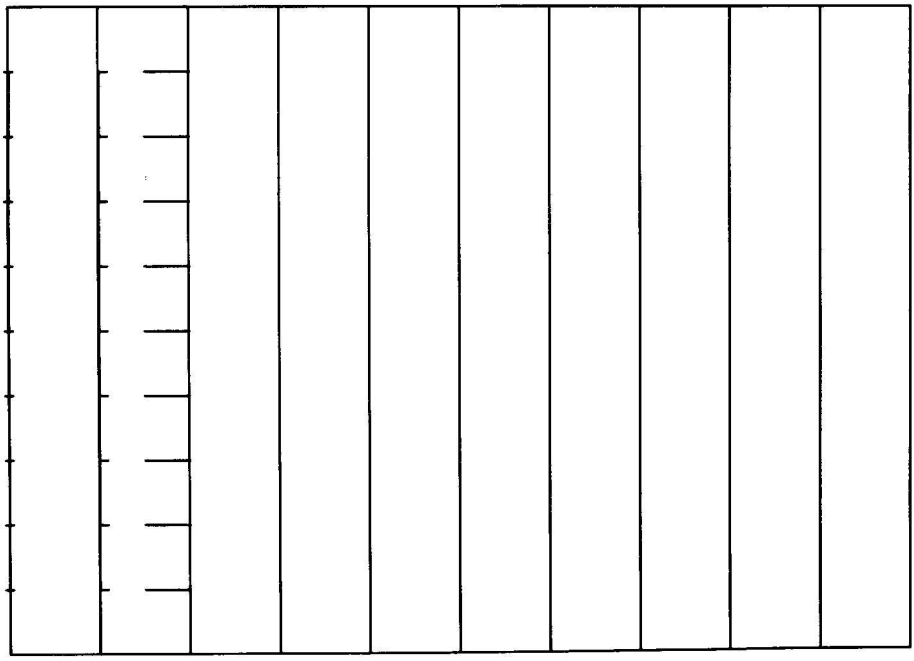
The down and left tick length, tn, determines the length of the downward portion of the tick marks drawn along the X-axis and the left-side portion of the tick marks drawn along the Y-axis, taking P1 as the lower-left corner.

The values specified by parameters tp and tn are a percentage of the vertical scale length ($P2_y - P1_y$) when used with the XT instruction, and a percentage of the horizontal scale length ($P2_x - P1_x$) when used with the YT instruction. Note the actual tick length is a function of the scaling established by P1 and P2, and the length of ticks on the X- and Y-axes will be different even if the same tick length percentage value is specified for both XT and YT, unless the area defined by P1 and P2 is square.

The plotter, when initialized, automatically sets the tick length values to 0.5% of the scaling lengths ($P2_y - P1_y$) and ($P2_x - P1_x$). A TL command with no parameters will default to the same values. A TL command with only one parameter specifies the length of tp, and tn will be zero. A negative tp parameter will draw a negative tick just as would be drawn by a tn with a positive parameter. Likewise, a negative tn parameter will draw a positive tick. Use of negative parameters is not recommended both because the results are more difficult to visualize and programs with negative parameters will not be compatible with other HP plotters. A TL command remains in effect until another TL command with valid parameters is executed or an IN or DF instruction is executed.

The following example draws both tick marks and grid lines. The grid lines are a result of specifying 100% tick length. The horizontal tick marks on the left-most grid line are drawn using the default tp,tn. The tick marks on the second grid line have a positive tick length of 1% and no negative tick. The tick marks on the third grid line have no positive tick and a negative tick length of 5%. Note that these last tick marks are drawn by the YT instruction even though the PU instruction is in effect. However, the moves to the next tick location are made with the pen up, and hence, the grid line is not retraced. A reduced version of the plot follows.

```
1 PRINTER IS 705,80
10 PRINT "IN;PA300,279;SP2;PD;TL100;XT;"
20 FOR I=1 TO 10
30 PRINT "PR1000,0;XT;"
40 NEXT I
50 PRINT "TL;PU;PA300,279;PD"
60 GOSUB 1000
70 PRINT "TL1,0;PU;PA1300,279;PD;"
80 GOSUB 1000
90 PRINT "TL0,5;PU;PA2300,279;"
100 GOSUB 1000
110 PRINT "PA300,7479;TL100;YT;PU;SP0;"
120 STOP
1000 ! SUBROUTINE TO DRAW TICKS
1010 FOR J=1 TO 9
1020 PRINT "PR0,720;YT;"
1030 NEXT J
1040 RETURN
1050 END
```



The Symbol Mode Instruction, SM

DESCRIPTION The symbol mode instruction, SM, is used with PA and PR commands, and provides the means to draw a single character which is centered at the end of each vector.

USES Symbol mode plotting can be used to draw a specified character at each data point and thus to create scattergrams, geometric drawings, or multiple-line graphs where lines are easy to differentiate.

SYNTAX *SM* c (terminator)
or
SM (terminator)

EXPLANATION An SM command without parameters turns off symbol mode. When a parameter is present, it is limited to a single character, which must be one of the printing characters of the character set currently selected.

After an SM command has been executed, subsequent PA and PR commands function as described in the previous chapter, except that the specified symbol mode character is drawn at the end of each vector and is centered on the plotted point. (A character drawn at a point using the label command, LB, would not be centered on the point.) Drawing of the character is independent of the current pen state (up or down); the character is always drawn at each point specified in the PA and PR command.

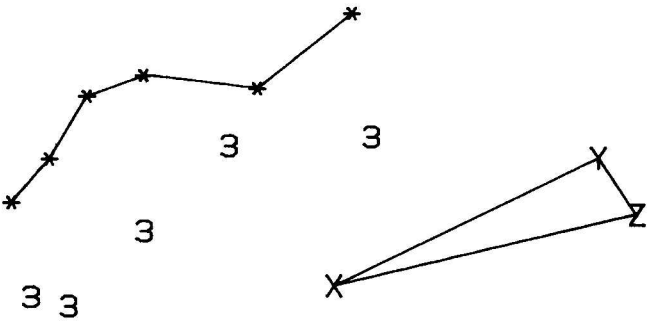
The character is drawn according to the character set selected when the SM command is executed. The character does not change even if a new set is selected. An SM command remains in effect until another valid SM command is executed or an IN or DF command is executed. The size (SI and SR), slant (SL), and direction (DI and DR) commands affect the character drawn.

An SM command can specify any printing character (decimal values 33 through 127). The semicolon (decimal value 59) is used only to cancel symbol mode (SM;) and cannot be selected as the symbol to be drawn at the endpoint of each vector. Specifying a space (decimal value 32) or any control character also cancels symbol mode.

The following example shows symbol mode plotting with the pen up and the pen down as might be used in line graphs, geometric drawings, and scattergrams.

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

Plot showing symbol mode:



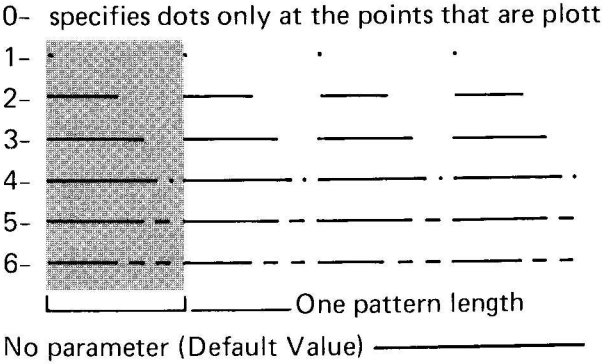
The Line Type Instruction, LT

DESCRIPTION The line type instruction, LT, specifies the type of line that will be used with PA and PR commands.

USES This instruction can be used with PA and PR commands to draw dashed or dotted lines. This facilitates trace differentiation on multiple-line graphs and enables emphasis or deemphasis of plotted lines or grids. One line type causes only dots to be plotted at each data point.

SYNTAX *LT* pattern number (,pattern length) (terminator)
or
LT (terminator)

EXPLANATION Shown below are the line patterns and their pattern numbers.



The shaded portion of each of the line patterns above is one complete segment of the pattern.

The pattern number parameter is in decimal format but is truncated to an integer. This parameter should be between 0 and 6; a parameter in this range sets the line type as shown in the preceding illustration. A parameter in the range 7 to 127.9999 is ignored; the line type does not change and no error is set. A parameter 128 or greater sets error 3 and the line type does not change. A negative parameter between 0 and −128 defaults to a solid line type and no error is set. A negative parameter less than −128 sets error 3 and the line type does not change.

When the first parameter is between 0 and 127.9999, the second parameter is used. This optional pattern length parameter is in decimal format. Both integer and fractional parts are used. This 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. When this parameter is positive and less than 127.9999, the pattern length is set to this length. When this parameter is negative or is greater than or equal to 128, the previous pattern length is used and error 3 is set. If a pattern length parameter is not specified, a length of 4% is used.

NOTE: If a vector ends in the pen-up portion of the pattern, a pen down command, PD, will not physically put the pen down until the next vector command is executed and the pen has moved so it is in a pen-down portion of a pattern segment. The pen up command clears the carry-over portion of a pattern segment. ■

Notes

Chapter 5
Labeling

What You'll Learn in This Chapter

In this chapter you will learn about character sets and labels used to create effective annotated graphics. You will learn how to designate and select character sets, how to use the label instruction with both constant and variable parameters, and how to set the size, slant, and direction of labels. Character spacing, moving the pen any number of character widths and/or lines, and designing your own characters will also be discussed.

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
DT The Define Terminator Instruction
LB The Label Instruction
DI The Absolute Direction Instruction
DR The Relative Direction Instruction
CP The Character Plot Instruction
SI The Absolute Character Size Instruction
SR The Relative Character Size Instruction
SL The Character Slant Instruction
*UC The User Defined Character Instruction

Terms You Should Understand

Label Terminator — the final character in every label string; it takes the plotter out of label mode so that characters are no longer drawn but are again interpreted as HP-GL instructions and parameters. Its default value is the ASCII character ETX (decimal equivalent 3), but it may be redefined using the DT instruction.

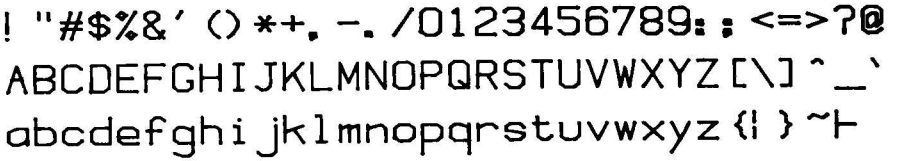
Character Space Field — the space occupied by a single character, together with the space between it and the next character and the space above the character which separates it from the previous text line.

*Not available with Option 003.

Plotter Character Sets

The plotter has the capability of lettering with any of five internal character sets. Each of the character sets has identical upper- and lowercase alphabetic characters and identical numerals. The symbols and punctuation marks vary from set to set, making annotation in several languages possible. The plotter, when initialized, automatically sets both the standard and alternate sets to ASCII character set 0 which follows:

CHARACTER SET 0



Some examples of annotation in foreign languages are found below. Notice that the label string in the HP-GL label command shows the character in the character set of the keyboard on which the command is entered or uses the CHR$ function if that ASCII character code is not available on the computer's keyboard.

```
"CS2;LB60 & DRU"&CHR$(123)&"BER"
```

60 & DRÜBER

```
"CS4;LB¿su compan"&CHR$(124)&"ia?"
```

¿su compañia?

```
"CS3;LB35-50 Å"&CHR$(124)&"R"
```

35-50 ÅR

Shown next are the symbols which vary from set to set. The plotter will perform an automatic backspace before drawing any of the shaded symbols. Therefore, when an accented letter is required, the letter should be entered first, followed by the accent.

Decimal Value	Set 0 Standard ASCII	Set 1 9825 Set	Set 2 French/German	Set 3 Scandinavian	Set 4 Spanish/Latin American
35	#	#	£	£	¿
39	'	'	´	'	´
91	[	[	[	Ø	[
92	\	ƒ	ç	Æ	i
93	]	]	]	ø	]
94	^	↑	^	æ	^
95	_	_	_	_	_
96	`	`	`	`	`
123	{	π	¨	¨	~
124	¦	⊢	°	°	~
125	}	→	¸	¸	~
126	~	~	'	°	~

The Designate Standard Character Set Instruction, CS

DESCRIPTION The designate standard character set instruction, CS, provides the means of designating one of the five character sets (0 through 4) as the standard character set.

USES The instruction can be used to change the standard character set to one with characters appropriate for your application. It is especially useful when labels are in a language other than English.

SYNTAX *CS* character set number (terminator)

EXPLANATION The character set number can be 0 through 4. The set designated by the CS instruction is used for all labeling operations when the standard set is selected by the SS instruction or by the control character shift-in (decimal equivalent 15) in a label string. Character set 0 is automatically designated as the standard character set whenever the plotter is initialized or set to default values.

A CS command executed while the standard set is selected will immediately change the character set used for labeling. CS commands

executed while the alternate set is selected will not change the set used for labeling until the standard set is selected.

A command CS with no parameters defaults to set 0. A CS command with an invalid first parameter will set an error condition (error 3), and the command will be ignored.

The Designate Alternate Character Set Instruction, CA

DESCRIPTION The designate alternate character set instruction, CA, provides the means of designating one of the five character sets (0 through 4) as the alternate character set.

USES The instruction can be used to provide an additional character set that can be easily accessed from a program, especially when a single label contains characters found in two different sets.

SYNTAX *CA* character set number (terminator)

EXPLANATION The character set number may be from 0 through 4. The set designated by the CA instruction is used for all labeling operations when the alternate set is selected by the SA instruction or by the control character shift-out (decimal equivalent 14) in a label string. Character set 0 is automatically designated as the alternate character set whenever the plotter is initialized or set to default values.

A CA command executed while the alternate set is selected will immediately change the character set used for labeling. CA commands executed while the standard set is selected will not change the set used for labeling until the alternate set is selected.

A command CA with no parameters defaults to set 0. A CA command with an invalid first parameter will set an error condition (error 3), and the command will be ignored.

The Select Standard Set Instruction, SS

DESCRIPTION The select standard set instruction, SS, provides the means of selecting the standard set designated by the CS instruction as the character set to be used for all labeling.

USES The command may be used to shift from the currently designated alternate character set to the currently designated standard character set so characters in another set may be accessed. Using the control character shift-in inside a label string is equivalent to executing this command.

SYNTAX *SS* (terminator)

EXPLANATION No parameters are used. Any parameters which follow the instruction are ignored and the standard set is selected. An alphabetic parameter will be interpreted as the first letter of the next mnemonic and may, therefore, cause an error 1 to occur after execution of the SS instruction.

The standard ASCII character set (set 0) is automatically selected when the plotter is first turned on, initialized, or set to default values. The standard set can be selected within a label command by sending the ASCII control character for shift-in (decimal equivalent 15).

The Select Alternate Set Instruction, SA

DESCRIPTION The select alternate set instruction, SA, provides the means of selecting the alternate set designated by the most recent CA instruction as the character set to be used for all labeling.

USES The command may be used to shift from the currently designated standard character set to the currently designated alternate character set to access characters in a second set. Sending the control character shift-out inside a label string is equivalent to executing this command.

SYNTAX *SA* (terminator)

EXPLANATION No parameters are used. Any parameters which follow the instruction are ignored and the alternate set is selected. An alphabetic parameter will be interpreted as the first letter of the next mnemonic and may, therefore, cause an error 1 to occur following execution of the SA instruction.

The command should be executed prior to executing a label statement whenever the alternate character set is to be used. The alternate set can be selected within a label command by sending the ASCII control character for shift-out (decimal equivalent 14). Shift-in and shift-out are particularly useful when a line of text must be composed with symbols from two character sets.

The following commands label using two different character sets where the underline is drawn with and without a backspace. The shift-out character is used to change from the standard to the alternate set.

```
"SP2;CSO;CR4;SS;LBS_E_T_O_%S_E_T_4_%"
```

S_E_T_O_SET4

The Define Terminator Instruction, DT

DESCRIPTION The define terminator instruction, DT, provides the means to specify the character to be used as the label terminator.

USES The command can be used to change the label terminator from its default value if ETX (decimal equivalent 3) cannot be used by your computer.

SYNTAX *DT* t (terminator) where t is the label terminator.

EXPLANATION The label mode can only be terminated by sending a label terminator at the end of the label character string. ASCII control characters (decimal equivalent 1 through 32) can be defined as label terminators and will not print when invoked, although the function normally performed by the character will be performed (i.e., LF will terminate a label but will also cause a line feed). ASCII characters with decimal equivalent values 33 through 127 can also be defined as the terminator, but the character will be printed at the end of the label character string. The ASCII control characters NULL (decimal equivalent 0) and ESC (decimal equivalent 27) cannot be used as label terminators. Also in the RS-232-C environment, ENQ (decimal equivalent 5) is not a valid terminator.

NOTE: A DT command with no parameter does not establish ETX as the default terminator, since the character immediately following the mnemonic DT is taken as a parameter. Only a DF or IN command or use of the ETX character itself as the instruction's parameter can be used to reestablish ETX as the label terminator. ■

The following examples of text in a label command demonstrate the use of the label terminator.

```
"IN;SP2;SC0,5000,0,5000;"
"PA0,4500;LBDefault control character ETX␃␊␅"
"LBterminates by performing end-␃␊␅"
"LBof-text function.␅"
"PA0,3900;DT#;LBPrinting characters terminate,␃␊#"
"LBbut are also printed.#"
"PA0,3400;DT␃;LBControl characters terminate␊␃"
"LBand perform their function.␃"
```

```
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.
```

The Label Instruction, LB

DESCRIPTION The label instruction, LB, provides the means to letter text, expressions, or string variables using the currently defined character set.

USES The label instruction can be used to annotate graphs or create text-only overhead transparencies.

SYNTAX *LB* c...c t

where t is the label terminator, either the default ETX character (decimal equivalent 3), or another character defined by the DT instruction.

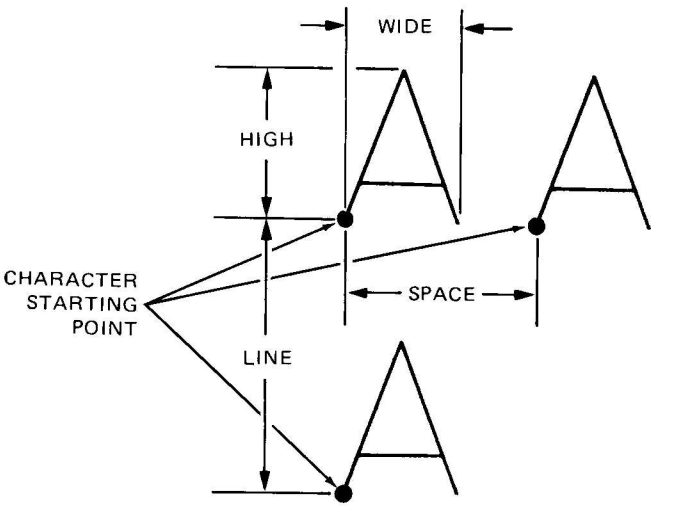
EXPLANATION All printing characters following the LB mnemonic are drawn using the currently selected character set. The set used is specified by the commands CA or CS and selected by the commands SA or SS, or the ASCII control characters shift-out or shift-in (decimal equivalent 14 and 15 respectively). If not specified, the default character set (set 0) is used.

The direction, size, and slant of the characters assume default values if not previously specified by DI, DR, SI, or SR commands.

The label mode can be terminated only by sending a label terminator at the end of the character string. Refer to The Define Terminator Instruction. (With an HP-IB interface, the bus commands interface clear IFC, device clear DCL, or selected device clear SDC will also terminate label mode. Refer to Bus Commands, Chapter 10.) Unless a label string is terminated, subsequent HP-GL commands will appear as labels in your plot.

The label begins at the current pen position. Before executing the LB command, the pen should be moved to the location where labeling is to begin using one of the plot commands (PA, PR, or a character plot command CP) or by front-panel controls. This establishes the lower-left

corner of the first character space and the carriage-return point. After lettering a character, the pen stops at the lower-left corner of the next character space as shown below. For a further explanation of character spacing, refer to Spacing Between Characters in this chapter.



When the plotter receives the character, carriage return, while in label mode, it returns to a defined carriage-return point. The carriage-return point usually reflects the pen's position when the preceding LB instruction was executed. The carriage-return point is updated to the current pen position whenever:

- one of the following instructions is executed: PA, PR, DI, DR, AA, AR, RO, DF, or IN.
- you use the front-panel CLEAR and RESET function keys or use the pen controls to move the pen to a new point.

Labeling with Variables

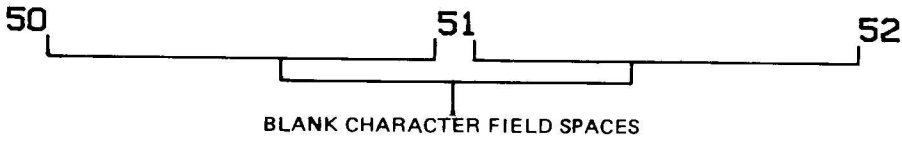
In some applications, it is desirable to label the plot using variables rather than literals to define the label string. Many different conventions are used in different computer languages and computers to define variable length and the character field format in which these variables will be printed. 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 first statement causes the plotter to label the value of X, X+1, and X+2. Blank spaces between the printed integers normally include space for the sign which may or may not be printed depending on your computer. The number of blank character-field spaces may vary with different computers.

```
"LB",X,X+1,X+2,"↖"
```



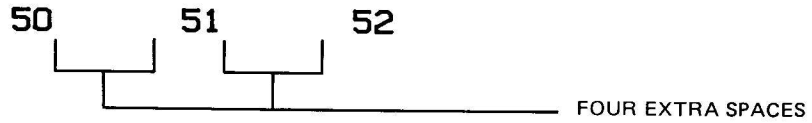
The following example illustrates the closer spacing achieved in BASIC when semicolons separate variables in labeling commands. The semicolons between the variables cause suppression of blank spaces. The space between the printed integers varies with different computers, but normally includes the sign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

```
"LB";X;X+1;X+2;"↖"
```

```
50 51 52
```

Any spaces required to fit into the context of the item being labeled must normally be sent enclosed in quotes. 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 quotes are sent between each variable, but the semicolon suppresses unwanted blank spaces.

```
"LB";X;"    ";X+1;"    ";X+2;"↖"
```



The Absolute Direction Instruction, DI

DESCRIPTION The absolute direction instruction, DI, specifies the direction in which characters are lettered.

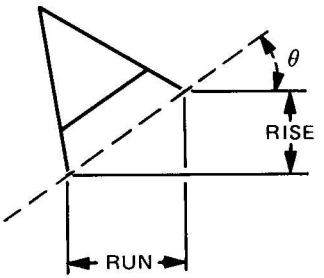
USES The instruction can be used to change the direction of labeling to a new absolute direction; by absolute we mean independent of P1,P2 settings. It is especially useful for labeling a Y-axis or labeling a vertical graph.

SYNTAX *DI* run, rise terminator
or
DI terminator

EXPLANATION Run and rise are in decimal format, 0 to ±127.9999, and specify the direction according to the relationship:

$$\theta = \tan^{-1}\left(\frac{\text{rise}}{\text{run}}\right)$$

where:



rise = SIN (θ)
run = COS (θ)

At least one parameter must be effectively nonzero, i.e., $|\ \geq 0.0004\ |$.

A DI command with a rise parameter of zero will produce horizontal labeling. A DI command with a run parameter of zero will produce vertical labeling.

A DI command with no parameters will default to the values DI 1,0 (horizontal). A DI command with only one or more than two parameters will set an error condition and the instruction will be ignored.

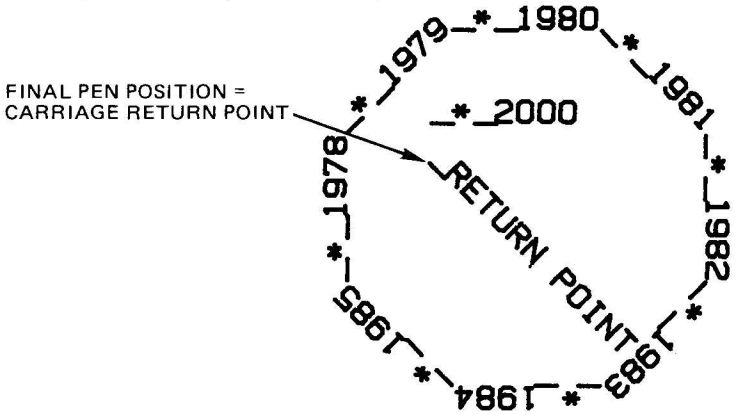
A change in the orientation of P1 and P2 will not affect the direction of labeling. A DI command remains in effect until another DI, DR, IN, or DF command is executed, or the plotter is initialized from the front panel.

A DI command updates the carriage-return point to the current pen position.

When the angle, θ, necessary to establish the desired label direction is known, the command DI $\cos\theta$, $\sin\theta$ can be used to establish label direction.

The following example labels the years 1978 through 1985, in a circular pattern starting with vertical labeling. The direction in which each year is labeled is changed by 45 degrees. Then the labels in the center are drawn to illustrate the use of cosine and sine values as parameters. The label _*_2000 contains both a carriage return and a line feed character before the label terminator, ETX, so the pen position at the end of that label is one line below the beginning of that label. The fact that DI commands update the carriage return point can be clearly seen 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 command.

```
"IN;SP2;PA1050,4450;"
"DI0,1;LB_*_1978↖ DI1,1;LB_*_1979↖"
"DI1,0;LB_*_1980↖ DI1,-1;LB_*_1981↖"
"DI0,-1;LB_*_1982↖ DI-1,-1;LB_*_1983↖"
"DI-1,0;LB_*_1984↖ DI-1,1;LB_*_1985↖"
"PA1500,5350;DI",COS(0),SIN(0);"LB_*_2000↵↖"
"DI",COS(-45);SIN(-45);"LB_RETURN POINT↵↖"
```



NOTE: Check the format of the COS and SIN functions on your computer, and change these accordingly. Also, check your computer documentation to see how your computer interprets angles. If angles are interpreted as radians, you need to change to degrees before using the COS and SIN functions. On the HP Series 80 computers, execute the BASIC statement DEG. ■

The Relative Direction Instruction, DR

DESCRIPTION The relative direction instruction, DR, specifies the direction in which characters are lettered.

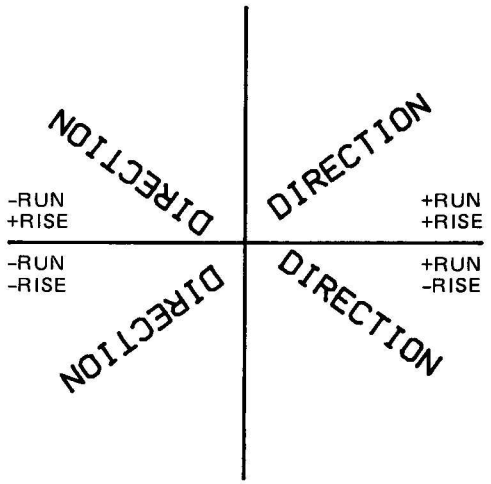
USES The instruction can be used to change the direction of lettering from its default direction, horizontal, to a direction which is relative to P1,P2 settings. It is useful when creating graphs which will be plotted in several sizes and you want labels to have the same relationship to the data on all plots.

SYNTAX *DR* run, rise terminator
or
DR terminator

EXPLANATION Run and rise are in decimal format, 0 to ±127.9999, and specify the label direction according to the same relationship specified in The Absolute Direction Instruction, DI.

Run and rise specify a percentage of the algebraic distance between P1 and P2 where run is the desired percentage (−128 to 127.9999) of $P2_x - P1_x$, rise is the desired percentage (−128 to 127.9999) of $P2_y - P1_y$, and P1 and P2 are the scaling points.

If you imagine the current pen position to be the origin, the sign of the parameters determines in which quadrant the lettering will be. In the example below, rise and run assume all combinations of ±1 with default P1 and P2.



A change in P1 or P2 will affect the direction of lettering. Refer to the section Parameter Interaction in Labeling Commands.

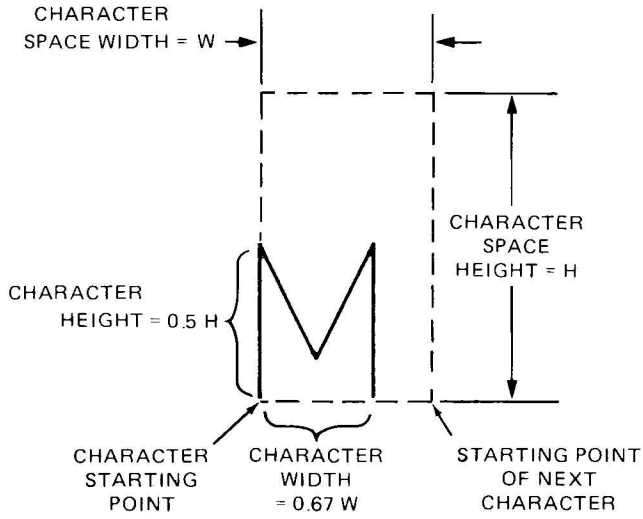
A DR command remains in effect until another DR or DI command or an IN or DF command or front-panel initialization is executed.

A DR command with no parameters will default to the values DR 1 , 0 (horizontal).

Specifying both parameters as zero will set error 3, and having only one or more than two parameters will set error 2. The plotter will ignore such instructions.

Spacing Between Characters

Character spacing and line spacing are functions of character size. In the diagram below, you can see the relative position of a character, in this case M, within the character space. The character-space field is set indirectly by the SI command, since the character space height is twice the character's height and the character-space width is 1½ times the character's width. The space above and beside a drawn character becomes the spacing between lines and characters. The character space is illustrated below.



When you specify the height of a character in an SI or SR command, however, you should specify the character height, not the height of a character space.

The Character Plot Instruction, CP

DESCRIPTION The character plot instruction, CP, moves the pen the specified number of character-space fields.

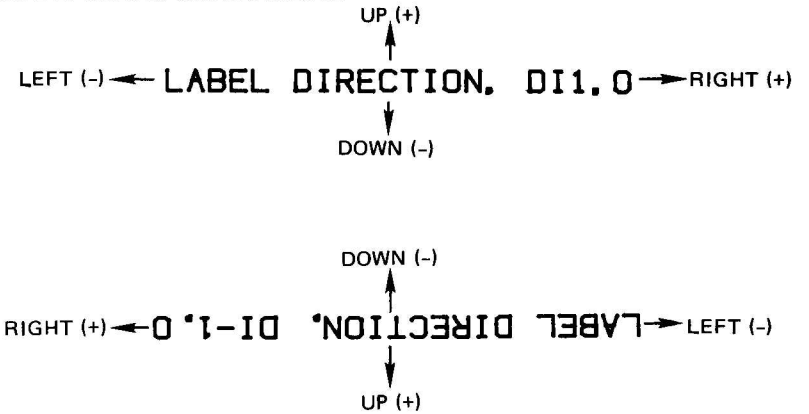
USES The instruction can be used to move the pen any number of character spaces or lines from a point on the plotting surface, to align with a left-hand margin, or to center or right-justify a label. Thus, the

label can be moved slightly above or below a line, spaces or lines can be inserted in text, or labels can be centered.

SYNTAX *CP* # of character-space-field widths, # of character-space field heights terminator
or
CP terminator

EXPLANATION If no parameters are specified, a CP command performs a carriage return and line feed, moving one character-space-field height down and returning to the margin defined by the carriage-return point. The carriage-return point is the last point moved to using either a PA, PR, PU, or PD command or front panel controls, or the pen position at the last DI or DR command. Refer to The Label Instruction in this chapter.

When parameters are specified, the CP command moves the pen the specified number of character-space-field widths to the right (a positive value) or the left (a negative value). Note that right, left, up, and down are relative to the label direction, where a positive value means from P1 toward P2. This is shown below.

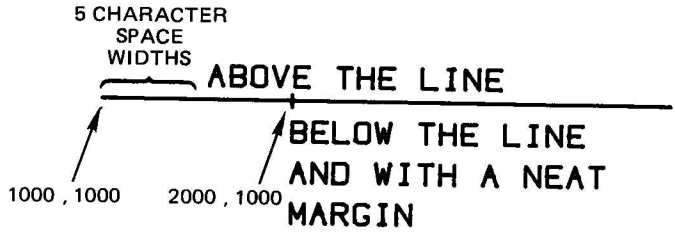


The pen's position (raised or lowered) does not change when a CP command is executed. The parameters must be ≥ -128 and $< +128$. However, since there are approximately 90 character-space-field widths and 40 character-space-field heights on the plotting surface, assuming default sizing, the effective parameter range that will keep the labels on the medium is considerably less, depending on the pen position at the given time.

The use of the CP command to produce lettering along a line, but not on top of it and alignment with a left-hand margin is illustrated in the following program. The CP command in the second line moves the label slightly above the line. The CP command in the third line moves the label slightly below the line and the CP command in the last line performs a carriage return, line feed to the margin established by the

plot command in the second line. Inserting carriage return and line feed characters directly into the label string in the third line causes the same effect as the CP; command in the last line. If the carriage return and line feed characters are available on your keyboard, you may prefer that method.

```
"DF;SP1;PA1000,1000PDPR3000,0PU;PR-3000,0;"
"CP5,.35;LBABOVE THE LINE↑ PA2000,1000;"
"XT;CP0,-.95;LBBELOW THE LINE↑↳AND WITH A NEAT↑"
"CP;LBMARGIN↑"
```



The Absolute Character Size Instruction, SI

DESCRIPTION The absolute character size instruction, SI, specifies the size of characters and symbols in centimetres.

USES The instruction can be used to change the character size from its default value or to another value and establish absolute character sizing in centimetres so character size is not dependent on the settings of P1 and P2.

SYNTAX *SI* width, height terminator
or
SI terminator

EXPLANATION If parameters are included, two parameters are required, width and height. The defined width and height are interpreted as centimetres, must be in decimal format, and may have any value between −128 and 127.9999. An SI command with no parameters will default to the values 0.19 for width and 0.27 for height.

An SI command remains in effect until another valid SI or SR command is executed or the plotter is initialized or set to default conditions. An SI command which sets an error condition is ignored and the character size does not change.

The following example letters the plotter's model number, 7470A, at the specified width of 1 cm and height of 1.5 cm.

"SI1,1.5;LB7470A"



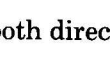
Negative SI parameters will produce mirror images of labels. A negative SI width parameter will mirror labels in the right-to-left direction.

COMMAND	RESULTING LABEL
"SI-.35,.6;LBHP"	

A negative height parameter will mirror labels in the top-to-bottom direction.

COMMAND	RESULTING LABEL
"SI.35,-.6;LBHP"	

Two negative SI parameters will mirror the label in both directions and the label will appear to be rotated 180 degrees.

COMMAND	RESULTING LABEL
"SI-.35,-.6;LBHP"	

For further information on the effects of negative parameters, refer to the section Parameter Interaction in Labeling Commands later in this chapter.

In order to produce legible characters, parameters should be greater than 0.1. Parameter values above 18 allow a maximum of one character to be drawn on the paper.

The Relative Character Size Instruction, SR

DESCRIPTION The relative character size instruction, SR, specifies the size of characters and symbols as a percentage of the distance between scaling points P1 and P2.

USES The instruction can be used to define character size relative to the distance between P1 and P2 so that if the P1,P2 distance changes, character size will adjust to occupy the same "relative" amount of space.

SYNTAX *SR* width, height terminator
or
SR terminator

EXPLANATION If parameters are included, two parameters are required, width and height. The defined width and height are interpreted as a percentage of the algebraic distance between the X- or Y-coordinates of P1 and P2. The parameters are in decimal format and may have any value between −128 and 127.9999. An SR command with no parameters will default to the values 0.75 for width and 1.5 for height, which, when P1 and P2 are at default values, produces letters the same size as an SI command without parameters.

An SR command remains in effect until another valid SI or SR command is executed or the plotter is initialized or set to default conditions. An SR command which sets an error condition is ignored and the character size does not change.

The following example shows how changes in P1 and P2 affect labels drawn while an SR command is in effect. The upper label is written with default character size. Then P1 and P2 are changed to define a square area with 6000-plotter-unit sides. A new label is drawn. Next a new SR command is executed with both width and height parameters set to three percent. Because the area established by P1 and P2 is square, equal parameters create square letters. With default P1 and P2 settings, equal parameters do not create square letters.

```
"IN;SP1;PA100,7000;LBDEFAULT SIZE↑"
"IP 1000,1000,7000,7000;PA100,6500;"
"LBNEW P1 AND P2 CHANGE LABEL SIZE↑ SR3,3;"
"PA100,6000;LBNEW SR COMMAND↑ ↳CHANGES LABEL SIZE↑"
```

DEFAULT SIZE

NEW P1 AND P2 CHANGE LABEL SIZE

NEW SR COMMAND
CHANGES LABEL SIZE

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 Commands for more information on mirroring.

With default P1 and P2, the useful range of width and height parameters which produces legible characters and a label of suitable length is 0.6 to 5.

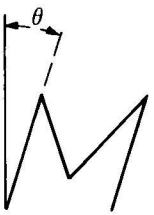
The Character Slant Instruction, SL

DESCRIPTION The character slant instruction, SL, specifies the slant with which characters are lettered.

USES The instruction may be used to create slanted text, particularly for emphasis, or to reestablish upright labeling after an SL command with parameters has been in effect.

SYNTAX *SL* tan θ (terminator)
or
SL (terminator)

EXPLANATION The instruction may be used with or without parameters. When parameters are included, the first parameter is interpreted as the tangent of the angle from vertical as shown below. Parameters following the first parameter are ignored. An SL command without parameters defaults to the same value as SL0 and labels are not slanted.



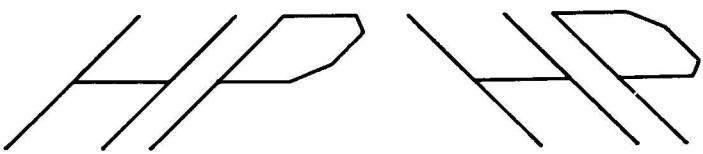


The useful parameter range is ±0.05 to ±2 when using default-size characters and up to ±3.5 for large letters.

An SL command remains in effect until an IN, DF or new SL command is received or the plotter is initialized from the front panel.

The following example letters HP at a slant of +45 degrees and −45 degrees.

```
"DF;SP1;SI1.3,1.8;PA3000,6000;"
"SL1;LBHP↑"
"SL-1;PR1300,0;LBHP↑"
```



The User Defined Character Instruction, UC

DESCRIPTION The user defined character instruction, UC, provides the means to draw characters of your own design. It is not included in the instruction set of the 7470 plotter with an HP-IL interface.

USES This instruction can be used to create symbols not included in the plotter's character sets, to draw logos, or to create your own character fonts.

SYNTAX *UC* (pen control,) X-increment, Y-increment,(pen control,) (X-increment,Y-increment,)...,..., terminator

or

UC terminator

EXPLANATION The instruction is treated as a NOP instruction on a plotter with an HP-IL interface (refer to Appendix C).

The following paragraphs apply to plotters with either an HP-IB or RS-232-C interface.

Each segment of the character is drawn on a character grid according to the three types of parameters in the command.

A grid is established on each character-space field by dividing it into six horizontal units and 16 vertical units. The size of the character-space field and, hence, the grid unit is set by the current size command. The size of the character-space field and thus the grid is always twice the current character height and 1½ times the current character width. In order to draw a user defined character the same size as a character drawn with a label command, the user defined character must be designed in the lower-left corner of the grid with a width of four grid units and a height of eight grid units.

The three types of parameters are described below.

The X- and Y-increments should appear in pairs and must be greater than −99 and less than +99. They specify, in decimal format, the number of X- or Y-grid units that the pen will move horizontally or vertically from the current pen position. The parameters need not be integers; fractional portions are used. Positive X-increment parameters move the pen in the direction of labeling, i.e., to the right with default label direction, and positive Y-increment parameters move the pen up with default label direction. Negative parameters move the pen in the opposite direction. Unmatched X,Y increments are discarded, error 2 is set, and the rest of the character is drawn.

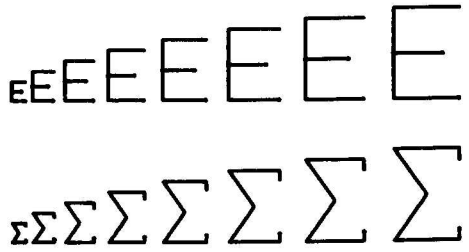
Pen control parameters must be less than or equal to −99 or greater than or equal to +99. A positive pen control parameter lowers the pen; a

negative pen control parameter raises the pen. Use of +99 and −99 is recommended. Once a pen down parameter has been sent, the pen will remain down for following X,Y increment moves until a negative pen parameter is received or the UC command is completed. Upon entry into a UC command the pen is raised. Each UC command must have at least one pen down parameter in order to draw anything. A UC command without a pen down will result in a pen movement of one character-space field horizontally. When a UC command is complete, the pen returns to its up/down status as set by PU or PD.

The position of the pen when the UC command is executed becomes the character origin point. The initial X,Y increment is relative to the character origin point and each subsequent move is relative to the last commanded pen position. Upon completion of the user defined character, the pen is automatically moved one character-space field to the right of the character origin point. This point becomes the current pen position and hence, the character origin point for the next character (if any).

The following example generates a Σ symbol which is the same size as an uppercase letter. For comparison, an "E" is drawn with the label command. The example shows how size commands affect both user defined characters and labeled characters. The HP-GL commands appear in quotation marks in the BASIC PRINT statements. Other BASIC statements, FOR and NEXT, are included in this example.

```
PRINT "IN;SP2;PA1000,1000;"
FOR A=.19 TO .89 STEP .1
PRINT "SI",A,A*1.4
PRINT "UC4,7,99,0,1,-4,0,2,-4,-2,-4,4,0,0,1;"
NEXT A
PRINT "PA1000,1750;"
FOR B=.19 TO .89 STEP .1
PRINT "SI",B,B*1.4
PRINT "LBE↖"
NEXT B
```



User defined characters need not fit into a single character-space field. In the next example, the user defined character takes up more than one character space. Since this character is to be followed by a label, a CP command must be added to move the current pen position beyond the limits of the user defined character. The reference point for parameters

of CP instructions is the pen position at the completion of the user defined character, one character-space field to the right of the origin of the user defined character.

```
"SP1;PA1000,5000;SI.25,.4"
"UC0,4,99,1.75,0,1.5,4,3,-8,3,8,3,-8,3,8,3,-8,1.5,4,1.75,0;"
"CP3.25,0;LB1000 ohms↖"
```



User defined characters are drawn using the current character size, slant, and direction. It is also possible to change the size of a user defined character by changing each X- or Y-increment parameter by a constant multiple. Send the following commands to the plotter. The resistor drawn will be twice the size of the resistor drawn in the last example.

```
"SP1;PA1000,4500;SI.25,.4"
"UC0,8,99,3.5,0,3,8,6,-16,6,16,6,-16,6,16,6,-16,3,8,3.5,0;"
```



Parameter Interaction in Labeling Commands

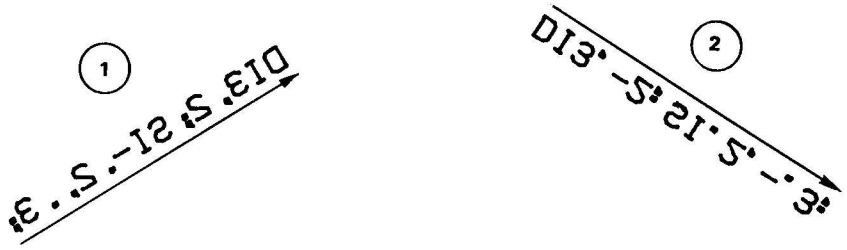
There are three factors which interact and affect the direction and mirroring of labels; the label direction as specified by DI or DR commands or default direction, the sign of the parameters for the size commands SI or SR, and the relative positions of P1 and P2. These interactions are complex. This 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 commands which cause the direction, size, and mirroring of the label. All descriptions are in terms of the standard X,Y coordinate system. 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, is always used. During the course of the illustrations, P1 and P2 are assigned to opposite corners of this rectangle in all possible ways. The values used for X-coordinates of P1 and P2 are 250 and 10 250; the values used for the Y-coordinates of P1 and P2 are 279 and 7479.

Use of DI and SI

When DI and SI commands are used together, the DI command establishes the label's direction and the SI command establishes its size. The direction serves as the axis along and about which labels (written with negative SI parameters) are mirrored. Positions of P1 and P2 do not affect the labels. Refer to The Absolute Direction Instruction, DI, and The Absolute Size Instruction, SI.

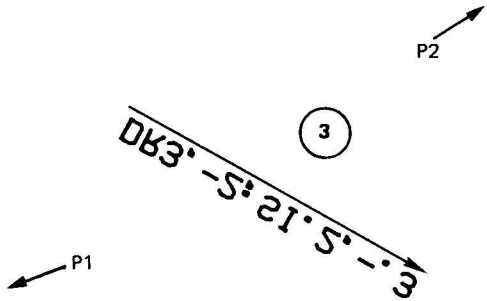
Two examples of mirrored labels are shown below. In the first example, the DI parameters 3,2 place the directional line in the first quadrant. The negative width parameter of the SI command mirrors the label in the right-to-left direction. In the second example, the DI parameters 3,−2 place the directional line in the fourth quadrant. The negative height parameter of the SI instruction mirrors the label top-to-bottom.



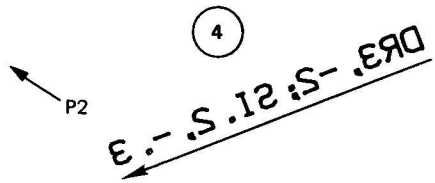
Use of DR and SI

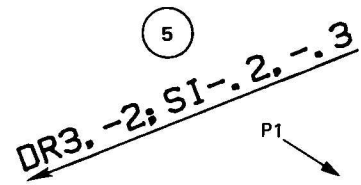
When DR and SI commands are used together, the label size is determined by the SI command and does not change with changes in the settings of P1 and P2. However, changes in the settings of P1 and P2 will affect the label direction. The algebraic differences ($P2_x - P1_x$) and ($P2_y - P1_y$) are multiplied by the run and rise parameters of the DR command. 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 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 becomes the upper-left corner. Now ($P2_x - P1_x$) is negative. The DR command as given is DR3,−2; the run parameter of the DR instruction is multiplied by −1 and the effective DR command becomes DR−3,−2 placing the directional line in the third quadrant. The negative SI height 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.



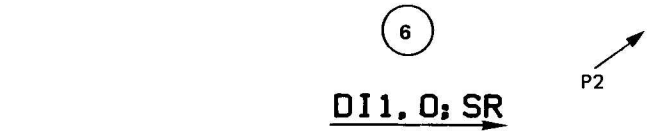


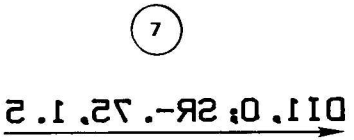
Use of DI and SR

When the DI command is used with SR, only the DI command 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

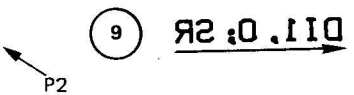
settings. In example 6, the SR; command is the same as SR.75,1.5. Since the parameters are positive, there is no mirroring. In example 7, the negative width parameter causes mirroring right-to-left. In example 8, the negative height parameter causes mirroring top-to-bottom.

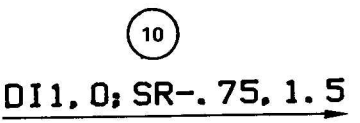






In the next three illustrations, P1 and P2 have been changed so P1 is lower right and P2 is upper left. Hence ($P2_x - P1_x$) is negative and anything with a positive SR width parameter is mirrored right-to-left, e.g., illustrations 9 and 11. The effect of the negative width parameter in illustration 10 is cancelled by the negative difference ($P2_x - P1_x$).

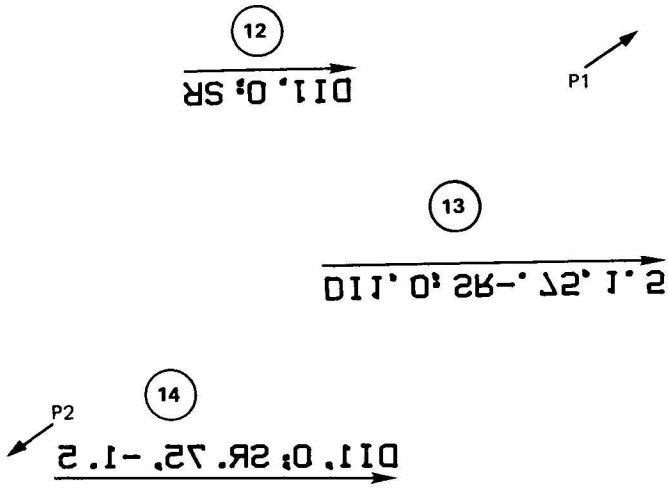






In the next illustrations, P1 and P2 have both been flipped so P1 is upper right and P2 is lower left. Now any positive parameter causes

mirroring and any negative parameter cancels mirroring. This can be seen in examples 12, 13, and 14.



Use of DR and SR

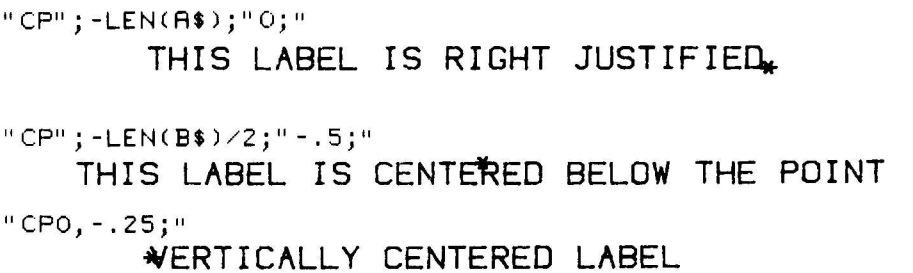
When the DR and SR instructions are used together, interactions are most complex. Using only standard settings of P1 and P2, where P1 is the lower-left corner and P2 is the upper-right corner, will make it easier for you to establish the direction and mirroring of labels you desire. DR parameters interact with the albegraic 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 differences are important. A negative sign in either the parameter or the distance will affect both DR and SR commands. Having both parameter and distance either positive or negative will cause standard direction or no mirroring.

When drawing labels, you often wish to position them precisely in relation to a specific point. Unless positioned differently by the programmer, labels are written beginning at the current pen position which marks the baseline of the label.

The following BASIC program illustrates various ways to center labels. The program uses the BASIC command LEN($) to find the length of the string. This length is used to determine horizontal adjustments, i.e., how many character-space widths the pen must be moved in order to achieve the desired positioning. Vertical moves are in terms of character-space heights. Since an uppercase letter is half the height of a character space, a vertical movement of one-quarter character space down will center uppercase letters on the point; notice the parameter is negative. A parameter of −0.5 will cause the top of uppercase letters to be level with the point.

Symbol mode plotting, with an * as the symbol, has been used here to show pen position at the start of the label command. The character plot instruction which positions the label is shown above each label.

```
10   DIM A$[40],B$[40],C$[40]
20   A$="THIS LABEL IS RIGHT JUSTIFIED"
30   PRINT "SP1;SM*;PA6000,5500;PDPU;"
40   PRINT "CP";-LEN(A$);"0;LB";A$;"^"
50   B$="THIS LABEL IS CENTERED BELOW THE POINT"
60   PRINT "PA4500,5000;PDPU;"
70   PRINT "CP";-LEN(B$)/2;"-.5;LB";B$;"^"
80   C$="VERTICALLY CENTERED LABEL"
90   PRINT "PA2750,4500;PDPU;"
100  PRINT "CP0,-.25;LB";C$;"^"
110  END
```



Chapter 6
Digitizing

What You'll Learn in This Chapter

The plotter can be used as a digitizer as well as a plotter. Digitizing consists of moving the pen or digitizing sight to a point on the plotting surface, entering the point, and sending the coordinates of that point to the computer. This chapter describes the three instructions used in digitizing, and contains a discussion of the steps required by a computer program for digitizing; sample programs are also included. Included in the discussion are three different methods of assuring that a point has been entered. The method you will use will depend on your application and your interface (HP-IB, HP-IL, or RS-232-C).

HP-GL Instructions Covered

DP The Digitize Point Instruction
DC The Digitize Clear Instruction
OD The Output Digitized Point and Pen Status Instruction

Terms You Should Understand

Digitizing — converting information, in this case pen position and up/down status, to digital information so that it can be understood by the computer.

Output Terminator — the character or characters sent by the plotter at the end of the response to an output command. It is interface-dependent.

Preparing Your Plotter for Use as a Digitizer

A plotter with an HP-IB interface must be set to an address less than 31 because the plotter cannot send the coordinates of a digitized point to the computer when it is in listen-only mode.

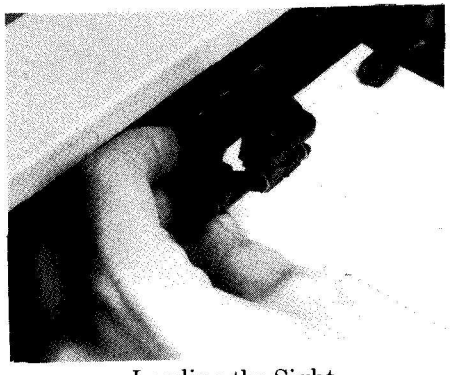
Use of a digitizing sight, available as an accessory with the 7470, is recommended. The sight should be loaded manually into the pen holder itself. It may be inserted into the pen holder from either side. Place the *flange on the digitizing sight* on top of the arm of the pen holder. The top of the sight will just clear the top of the pen holder. Push the sight gently into the pen holder; it will snap into place.

CAUTION

The sight should not be stored in a pen stall; do not store using front panel buttons or an SP command. Remove the sight from the pen holder before raising the **PAPER LOAD** lever since the sight would be stored automatically when the lever is raised.

To remove the sight from the pen holder, pull either arm of the pen holder forward and push the sight out of the pen holder.

The sight is used in the pen down position.



Loading the Sight



Unloading the Sight

The Digitize Point Instruction, DP

DESCRIPTION The digitize point instruction, DP, provides the means to digitize points on the plotter.

USES This instruction can be used to input data for a graphics program or obtain the coordinates of a point or points on the plot.

SYNTAX *DP* (terminator)

EXPLANATION No parameters are used. The instruction will execute even if no terminator is received.

When the DP command is received, automatic pen lift is suppressed and the plotter is ready to have a digitized point entered by pressing **ENTER** on the front panel.

When **ENTER** is pressed, the X- and Y-coordinates of that point and pen up/down status are stored for retrieval by the OD command. Pressing **ENTER** sets bit position 2 of the status byte, indicating a digitized point is available for output.

After **ENTER** has been pressed, automatic pen lift is reactivated.

The Digitize Clear Instruction, DC

DESCRIPTION The digitize clear instruction, DC, provides a means to terminate digitize mode.

USES This instruction can be used to terminate digitize mode without entering a point. If you are using an interrupt routine in a digitizing program to branch to some other plotting function, you could use DC to clear digitize mode immediately after branching.

SYNTAX *DC* (terminator)

EXPLANATION No parameters are used. The instruction will execute even if no terminator is received.

When the DC command is received, digitize mode is terminated. Automatic pen lift is reactivated.

The Output Digitized Point and Pen Status Instruction, OD

DESCRIPTION The output digitized point and pen status instruction, OD, is used to output the X- and Y-coordinates and pen up/down status associated with the last digitized point.

USES This instruction is used after DP and **ENTER** in all digitizing applications to return the coordinates of the digitized point to the computer.

SYNTAX *OD* (terminator)

EXPLANATION No parameters are used. The instruction will execute even if no terminator is received.

The timing of output depends on the plotter's interface (HP-IB, HP-IL, or RS-232-C). Refer to A Brief Word about Plotter Output in Chapter 7 for more information.

The pen position and status are output to the computer as integers in ASCII in the form:

X,Y,P [TERM]

where X is the X-coordinate of the digitized point in plotter units,
Y is the Y-coordinate of the digitized point in plotter units,
P is the pen status when the point was entered (0 = pen up, 1 = pen down), and
[TERM] is the output terminator for your system (refer to Chapter 7).

The ranges of the X- and Y-coordinates are the mechanical limits of the plotter as determined by the setting of the paper switch.

Upon receipt of the OD command by the plotter, bit position 2 of the output status byte is cleared.

Digitizing with the 7470

When using the plotter as a digitizer, it is important to ascertain that a point has been entered before an attempt is made to retrieve that point using the OD command. There are three methods for doing this.

Manual Method

The first method, which might be called the manual method, is easiest to understand. It is not efficient in applications where many points will be entered, or in an RS-232-C environment where the mainframe is not adjacent to the plotter or where human intervention in program execution is not possible. The steps in this method are as follows:

1. In a program, send a DP command to the plotter. Follow the DP command immediately with a statement that will cause the program to display or print a message prompting you to enter a point. Follow the prompt with a statement that will cause the program to pause until instructed to continue. The BASIC statement PAUSE will accomplish this.
2. Move the digitizing sight (pen) to the point to be entered, using front-panel buttons. Final positioning should be done with the sight (pen) down.
3. Press **ENTER** on the plotter's front panel. Now resume running of the program. This is done on HP desktop computers by pressing the key marked **CONTINUE** or **CONT**.

4. The program step following the pause will now be executed. The next steps of the program, in order, should be an OD command to the plotter, a read statement by the computer to read the X- and Y-coordinates and the pen status, a statement to remove the prompt (requesting you to enter a point) from the screen, and then steps to process the digitized data in the appropriate manner.

Using this method, there is no need to monitor the status byte because the program does not proceed to the OD command until the user enters a point and causes the program to resume.

A simpler procedure, using OA or OC instead of OD, can also be used. It omits the DP in step 1 and pressing **ENTER** in step 3. Using the shorter procedure with OC makes it possible to obtain coordinate values in user units. Refer to Chapter 7.

A short program to digitize a single point and display the coordinates and pen status is given below. The program is in BASIC for an HP-85 with an HP-IB interface. An I/O ROM is required in order to execute the ENTER statement to obtain the digitized point.

```
10 PRINTER IS 705,80
20 PRINT "DP;"
30 DISP "ENTER A POINT"
40 PAUSE
50 PRINT "OD;"
60 ENTER 705 ; X,Y,P
70 DISP X;Y;P
80 END
```

Monitoring the Status Byte

The second method can be used with any interface and is the only method of checking based on software that can be done in an RS-232-C environment. This 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, Chapter 7 for more information.

Monitoring bit position 2 can be done in a variety of ways depending on the commands available on the computer being used. If there are instructions to check bits directly, the third least significant bit (lsb) should be checked for the occurrence of a 1. If no bit operations are available, the status byte can be operated on arithmetically to check for the availability of a digitized point. Executing successive divisions of a number by two and checking for an odd or even integer answer is a common way of monitoring bits without converting the number to binary form. Either of the following sequences of BASIC instructions

will check the proper bit of the status byte. Insert as line 110 or line 1010 a suitable BASIC read statement to read the status byte into a variable called Status.

```
.
.
.
100 PRINT "OS;"
110                ! STORE STATUS BYTE IN Status
120 Status=INT(Status/2) !SHIFTS BITS RIGHT ONE POSITION
130 Status=INT(Status/2) !SHIFTS BITS RIGHT AGAIN
140 Status=Status MOD 2  !THIS RESULT IS 0 IF LSB NOT 1
150 IF Status=0 THEN 100
160 PRINT "OD;"          !SEND OD SINCE POINT AVAILABLE
.
.
.


.
.
.
1000 PRINT "OS;"
1010               ! STORE STATUS BYTE IN Status
1020 Status=INT(Status/4) !SHIFTS BITS RIGHT 2 POSITIONS
1030 IF Status=INT(Status/2)*2 THEN 1000  !1sb NOT 1
1040 PRINT "OD;"
.
.
.
```

On some HP computers with an I/O ROM, the following three lines are equivalent to lines 100 to 150 of the first program segment shown.

```
2000 PRINT "OS;"
2010          !THIS IS THE STATEMENT TO READ THE STATUS
2050 IF BIT(Status,2)=0 THEN 2000
```

In many applications, a large number of points need to be digitized. When the computer is used to monitor bit position 2, the points may or may not be processed immediately. In most applications, memory would be allocated for the total number of points to be digitized. A loop would be established to process the total number of points, calling the subroutine each time to check that a point had been entered. A complete BASIC program for an HP-85 with an HP-IB interface follows. This program prints out the 500 points after they all have been entered.

```
10 PRINTER IS 705,80
20 OPTION BASE 1
30 INTEGER X(500),Y(500),P(500)
40 FOR C=1 TO 500
50 PRINT "DP;"
60 DISP "ENTER POINT ";C
70 GOSUB 160
80 PRINT "OD;"
90 ENTER 705 ; X(C),Y(C),P(C)
100 NEXT C
110 PRINTER IS 2
120 FOR C=1 TO 500
130 PRINT X(C);Y(C);P(C)
140 NEXT C
150 STOP
160 ! Check SUBROUTINE
170 PRINT "OS;"
180 ENTER 705 ; S
190 S=INT(S/4)
200 IF S=INT(S/2)*2 THEN 170
210 RETURN
220 END
```

HP-IB Interrupts and Polling

A third method can be used by advanced programmers thoroughly familiar with the HP-IB interface, polling techniques, and interrupts. It should only be used when the 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 command, e.g., IM 223 , 4 , 0 ; to cause the plotter to generate an RQS (service request) when a digitized point is available. With an interrupt routine enabled for service requests, the computer can send a DP command 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 RQS, and program execution branches to the routine to process the digitized data. This routine could simply send an OD command 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 RQS. The status byte can be obtained by serial polling or simply by sending an OS command. Because interrupts and polling are highly machine-dependent and beyond the scope of this manual, no examples are given.

Notes

Chapter **7**

Obtaining Information from the Plotter

What You'll Learn in This Chapter

Up to this time we have mainly been concerned with sending information or data to the plotter. Sometimes, however, we want to know something about the plotter, its current pen position, its status, whether an error has occurred, or what capabilities the plotter has. In this chapter you will learn about most of the plotter's output instructions. The output P1 and P2 and output window instructions are discussed in Chapter 2 and the output digitized point instruction is discussed in Chapter 6. All other output instructions are discussed in this chapter. The timing of output depends on your interface (HP-IB, RS-232-C, or HP-IL). Before using the output instructions, you should have read the notes below and the appropriate interfacing chapter in this manual.

HP-GL Instructions Covered

OA The Output Actual Position and Pen Status Instruction
OC The Output Commanded Position 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

Terms You Should Understand

Output Terminator — denoted in this manual as [TERM] — the ASCII character or characters sent by the plotter at the end of a plotter response to an output command. With an HP-IB or HP-IL interface, the two characters, carriage return and line feed, are the output terminator. With an RS-232-C interface, the output terminator is a carriage return, unless modified by an ESC . M command.

A Brief Word about Plotter Output

There are slight differences in the timing of output when the plotter is used with the HP-IB, HP-IL, or RS-232-C interfaces. Read the paragraph below which pertains to your system.

Notes for an HP-IB User

When the 7470 has an HP-IB interface, the terminator for an output statement, denoted [TERM], is a carriage return followed by a line feed.

The output instructions in this chapter should not be used when the plotter is in listen-only mode since the plotter in listen-only mode cannot output anything. Output instructions will be ignored by the plotter so the computer will get no response to its read statement, and, typically, the program will halt.

A plotter with an HP-IB interface will respond only when the computer sends a read command (the plotter is instructed to talk). Therefore, a read statement should directly follow any output command. When a second output command is received before data from the first command has been read, the new data overwrites the old data and the old data is lost. Refer to Chapter 9 for more information.

Notes for an RS-232-C User

With an RS-232-C interface, the 7470's terminator for an output statement, denoted [TERM], is a carriage return, unless the terminator is modified by an ESC . M command. As soon as an output command has been parsed by the plotter, output occurs according to the handshake protocol established by the ESC . M and ESC . N commands. Use of turnaround delays, intercharacter delays, and an output initiator should be specified as necessary to assure that output will not be lost because the computer is not prepared to receive it. The information necessary to assure this should be contained in the documentation for your computer. Refer to Chapter 10 of this manual for more information.

Notes for an HP-IL User

When the 7470 has an HP-IL interface, the terminator for an output statement, denoted [TERM], is a carriage return followed by a line feed. A plotter with an HP-IL interface will only respond when it is instructed by the controller to talk. Therefore, a read statement should follow any output command so that the plotter can send the requested information. There are no special output timing considerations with HP-IL. This is because data are sent through the interface bit-serially; only one message can travel through the loop at a given time. Refer to Chapter 11 and your computer's documentation for more information.

The Output Actual Position and Pen Status Instruction, OA

DESCRIPTION The output actual position and pen status instruction, OA, is used to output the X- and Y-coordinates and pen status (up or down) associated with the actual pen position.

USES This instruction can be used to determine the pen's current position in plotter units. You might use that information to position a label or figure, or determine the parameters of some desired window.

SYNTAX *OA* (terminator)

EXPLANATION Output is always in plotter units.

No parameters are used. The instruction will execute even if no terminator is received.

The pen position and status are output to the computer as integers in ASCII in the form:

X,Y,P [TERM]

where X is always the X-coordinate in plotter units,
Y is always the Y-coordinate in plotter units,
P is the pen status (0 = pen up, 1 = pen down), and
[TERM] is the output terminator for the interface installed.

The ranges of the X- and Y-coordinates are the current mechanical limits determined by the setting of the paper switch.

US	A4
$0 \leqslant X \leqslant 10\,300$	$0 \leqslant X \leqslant 10\,900$
$0 \leqslant Y \leqslant 7650$	$0 \leqslant Y \leqslant 7650$

No positive sign is output.

The Output Commanded Position and Pen Status Instruction, OC

DESCRIPTION The output commanded position and pen status instruction, OC, is used to output the X- and Y-coordinates and pen status (up or down) associated with the last valid pen position command.

USES This instruction can be used 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 that information to position a label or figure, or determine the parameters of an instruction which moved the pen to the limits of some window.

SYNTAX *OC* (terminator)

EXPLANATION Output is in decimal format, in user units when scaling is in effect, and in plotter units when scaling is off.

No parameters are used. The instruction will execute even if no terminator is received.

The pen position and status are output to the computer as decimal numbers in ASCII in the form:

X,Y,P [TERM]

where X is always the X-coordinate in plotter units or user units,
Y is always the Y-coordinate in plotter units or user units,
P is the pen status (0 = pen up, 1 = pen down), and
[TERM] is the output terminator for the interface installed.

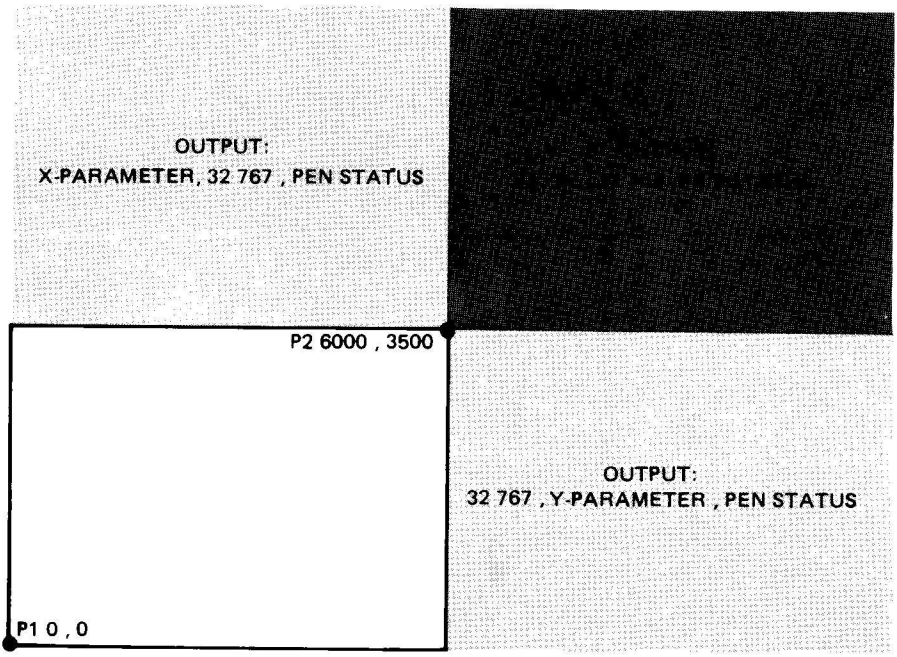
When scaling is off, X- and Y-coordinates are in plotter units. When scaling is on, X- and Y-coordinates are in user units. Ranges of the X-and Y-coordinates are −32 768 to 32 767 whether scaling is on or off.

NOTE: If you have an HP-IB or RS-232-C plotter that has the serial prefix number **2308A** or higher, or if you have an HP-IL plotter, output is in decimal format as described above. All HP-IB or RS-232-C plotters with a **lower** prefix serial number output integer parameters, as follows. When scaling is on, X- and Y-coordinates are always rounded to the nearest integer value. Thus, while plotting can occur to noninteger values, output of pen position can only be obtained to the nearest integer value. ■

When the commanded pen position is such that its user unit value would be less than −32 768 or greater than 32 767, the output may not represent the true pen position. If the plotter were scaled with the given instructions as shown in the following illustration, all points in the lightly shaded area will have one coordinate as 32 767, the largest number the plotter can output. All points in the darker shaded area will have both coordinates as 32 767.

Commands executed:

```
"IP 0,0,6000,3500;SC 0,32767,0,32767;"
```



The Output Error Instruction, OE

DESCRIPTION The output error instruction, OE, is used to output the decimal equivalent of the last HP-GL error (if any).

USES This instruction can be used to determine the type of the last error. It is useful when debugging programs or to determine if all data or instructions were accepted by the plotter.

SYNTAX *OE* (terminator)

EXPLANATION No parameters are used. The instruction will execute even if no terminator is received.

When an OE command is received, the plotter converts the last HP-GL error to a positive integer in ASCII, which is output in the form:

error number [TERM]

The error number is defined as follows:

Error Number	Meaning
0	No error
1	Instruction not recognized
2	Wrong number of parameters
3	Out-of-range parameters, or illegal character
4	Not used
5	Unknown character set
6	Position overflow
7	Not used
8	Vector received while pinch wheels raised

[TERM] is the output terminator for the interface installed.

In an HP-IB or an HP-IL system after the carriage return has been sent, and in an RS-232-C system after the output is complete, bit position 5 of the status byte is cleared (if set), and the **ERROR** LED (if lit) is turned off (unless there is an RS-232-C error which has not been cleared by an ESC . E command).

You should note that anytime the plotter receives an unpaired alphabetic character, error 1 will be set. Thus, an alphabetic parameter or three alphabetic characters in a row will generate error 1. When you encounter error 1, look for a misplaced alphabetic character.

Once your plotting programs are debugged, you may want to remove most output error instructions from your program to reduce your computer's I/O operations and maximize plotting speed.

The Output Factors Instruction, OF

DESCRIPTION The output factors instruction, OF, is used to output the number of plotter units per millimetre in each axis.

USES This instruction enables the plotter to be used with software which must know the size of a plotter unit.

SYNTAX *OF* (terminator)

EXPLANATION No parameters are used. The instruction will execute even if no terminator is received.

The plotter will always output the following:

40,40[TERM]

These factors indicate that there are approximately 40 plotter units per millimetre in the X-axis and in the Y-axis (0.025 mm/plotter unit). [TERM] is the output terminator for the interface installed.

The Output Identification Instruction, OI

DESCRIPTION The output identification instruction, OI, is used to output a plotter identifier.

USES This instruction is especially useful in a remote operating environment to determine which model plotter is on-line.

SYNTAX *OI* (terminator)

EXPLANATION No parameters are used. The instruction will execute even if no terminator is received.

The plotter will always output the following character string:

7470A [TERM]

[TERM] is the output terminator for the interface installed.

The Output Options Instruction, OO

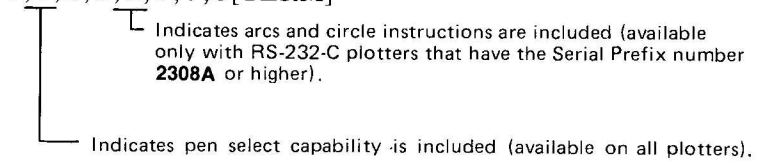
DESCRIPTION The output options instruction, OO, is used to output eight option parameters.

USES This instruction is especially useful in a remote operating environment to determine which options are available in the plotter which is on-line.

SYNTAX *OO* (terminator)

EXPLANATION No parameters are used. The instruction will execute even if no terminator is received.

The plotter will always output the appropriate combination of eight integers in ASCII, separated by commas. The options included in the plotter are indicated by a 1 as defined below.



[TERM] is the output terminator for the interface installed.

The Output Status Instruction, OS

DESCRIPTION The output status instruction, OS, is used to output the decimal equivalent of the status byte.

USES This instruction is useful in debugging operations and in digitizing applications.

SYNTAX *OS* (terminator)

EXPLANATION No parameters are used. The instruction will execute even if no terminator is received.

Upon receipt of the OS instruction, the internal eight-bit status byte is converted to an integer between 0 and 255. Output is in ASCII in the form:

status [TERM)

The status bits are defined as follows:

Bit Value	Bit Position	Meaning
1	0	Pen down.
2	1	P1 or P2 changed; cleared by reading output of OP in HP-IB or HP-IL system or by actual output of P1,P2 in RS-232-C system.
4	2	Digitized point available; cleared by reading digitized value in HP-IB or HP-IL system or by output of point in RS-232-C system.
8	3	Initialized; cleared by reading OS output in HP-IB or HP-IL system or by output of the status byte in RS-232-C system.
16	4	Ready for data; pinch wheels down.
32	5	Error; cleared by reading OE output in HP-IB or HP-IL system or by output of the error in RS-232-C system.
64	6	Require service message set (always 0 for OS; 0 or 1 for HP-IB serial poll).
128	7	Not used

Upon power up, the status is decimal 24, the sum of 8 (initialized) and 16 (ready for data). Upon output of the status byte after an OS command, bit position 3 is cleared.

Summary of Output Response Types

The following table shows the number and type of items in the response to each HP-GL output command. The table includes output commands explained in Chapters 2 and 6 as well as in this chapter. This table will be helpful when programming in languages such as FORTRAN which require you to specify the type of and number of digits in a variable.

Instruction	Number of Parameters Returned*	Type and Range
OA	3	integers, all $\leq$ 5 digits
OC**	3	decimals, all $\leq$ 11 digits
OD	3	integers, all $\leq$ 5 digits
OE	1	integer, 1 digit
OF	2	integers, 2 digits each
OI	1	5-character string
OO	8	integers, 1 digit each
OP	4	integers, 1st and 3rd $\leq$ 5 digits; 2nd and 4th $\leq$ 4 digits
OS	1	integer, $\leq$ 3 digits
OW	4	integers, 1st and 3rd $\leq$ 5 digits; 2nd and 4th $\leq$ 4 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.

If you have an HP-IB or RS-232-C plotter that has a serial prefix number *lower than* **2308A, OC parameters are output as integers. For more information, refer to the explanation of the OC instruction in this chapter.

Notes

Chapter 8
Putting the Commands to Work

What You'll Learn in This Chapter

In this chapter you'll learn how to put commands together to develop a plot. Previous programs have been purposely kept to a less-advanced level in order to clearly demonstrate the command usage. The following example is designed to show you how to integrate many commands into a complete program, how data might be handled, and how subroutines might be used to program a task which would be common to many plots and used in several programs.

This program draws a line graph, one of the most common types of plots. While this line graph shows sales data, line graphs 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 are applicable in almost any application.

A variety of allowable separators and terminators have been used in this program listing. In applications where it is important to minimize the number of characters sent over the interface, the spaces between commands and the semicolon preceding the next mnemonic could and should be omitted. In applications where compatibility with other HP plotters is important, a semicolon or a line feed should always be used as the terminator and parameters should be separated by commas. With RS-232-C plotters, use a semicolon; line feeds are not recognized as terminators.

Problem

Scale, draw, and label an X- and Y-axis in user units and plot 1981 sales by sales region. Use a different line type for each sales region and place a legend on the graph. The complete program is in the Listing section, following the Solution section.

Solution

Setup and Scaling

The first step is to set the plotter to known conditions, cancelling any parameters which may have been set in the previously run program. The IN or DF instruction may be used; IN resets P1 and P2; DF does not. IN is used here.

Next, a pen is selected (SP1) and the scaling for this plot is established. The parameters of the IP command determine the location of the scaling points, P1 and P2. In this graph, all data will be plotted within this P1,P2 area. The points have been chosen to allow room for labels, titles, and margins outside the P1,P2 rectangle. The scaling statement SC1,12,0,150; assigns user unit values to the scaling points. Since we are plotting one year's sales by month, we have scaled the X-axis (commonly representing time) from 1 to 12. The Y-axis is scaled in thousands from 0 to 150 so all sales data fall well inside the scaled 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 graph is scaled from 0 to 150, not 0 to 150 000 — the actual range of sales dollars. There are two reasons for this. First, the largest number accepted by the plotter is 32 767; our numbers are too large so we need to divide all data by at least 10. In this program, both labels and data will be stated in thousands. It is easier to interpret a scale marked with short labels. The eye need only read a maximum of three characters (150) instead of six (150 000). Thousands or millions of dollars are common scales for graphs.

Having established our scaling, we shall draw a frame for the data area. This is done by moving to the point 1,0 with the pen up, lowering the pen and drawing to the four corners 12,0;12,150;1,150; and 1,0. The coordinates are interpreted as absolute (instead of relative) moves since absolute plotting is established by the IN command. The first three program lines with HP-GL commands are:

```
20 PRINT "IN;SP1;IP1250,750,9250,6250;"
30 PRINT "SC1,12,0,150;"
40 PRINT "PU1,0 PD 12,0,12,150,1,150,1,0 PU"
```

NOTE: If compatibility with other 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 commands. In addition, the stricter syntax of other plotters would be required. ■

The Axes and Their Labels

We are now ready to draw and label the axes. The label size is set by the absolute size command SI .2 , .3 ; . This creates characters which are slightly larger than characters of default character size specified by the IN command. The tick length is established by the tick length command 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 in BASIC uses FOR...NEXT loops. First, we shall draw the X-axis. Let X range from 1 to 12 representing the 12 months for which we have data. In the loop we will do four things: move to the integer location on the axis, draw a tick mark, position the pen below the axis, and draw the label. Note that the X-parameter of the plot command is a variable. You will need to know how to send a variable between strings of fixed characters. The method will differ from computer to computer; consult your computer's documentation and Plotting with Variables in Chapter 3 of this manual. If you have an HP-IB or HP-IL plotter, refer also to Sending and Receiving Data in Chapter 9 or 11. The XT instruction draws a tick, whether the pen is up or down. The pen is up here so we do not draw the axis line again. You might want to use PD, drawing over the frame line if your want your axis line a bit darker, or you might want to redraw the axis again later with a wide pen.

There are several techniques used here to draw the alphabetic labels. First, so we can use a looping technique, we have placed 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 label and data from some file and properly label your graph for the data you were then plotting.) Secondly, we have used the CP instruction together with BASIC formatting (using semicolons to suppress extra characters between print fields) to center the label under the tick. The base of the tick mark is the pen position after the tick is drawn. By moving one-third character space back and one line down, the single character label is centered under the tick with enough space so it can be easily read. Finally, the axis title, CALENDAR MONTH, is centered and drawn under the axis.

The loop to draw the axis and the statements to set character and tick length and to label and title the X-axis are:

```
50 PRINT "SI.2,.3;TL1.5,0"
60 FOR X=1 TO 12
70 PRINT "PA";X;",0; XT;"
80 READ A$
90 PRINT "CP-.33,-1;LB";A$;"ETX"
100 NEXT X
110 PRINT "PA6.5,0;CP-7,-2.5; LBCALENDAR MONTHETX"
400 DATA "J","F","M","A","M","J","J","A","S","O","N","D"
```

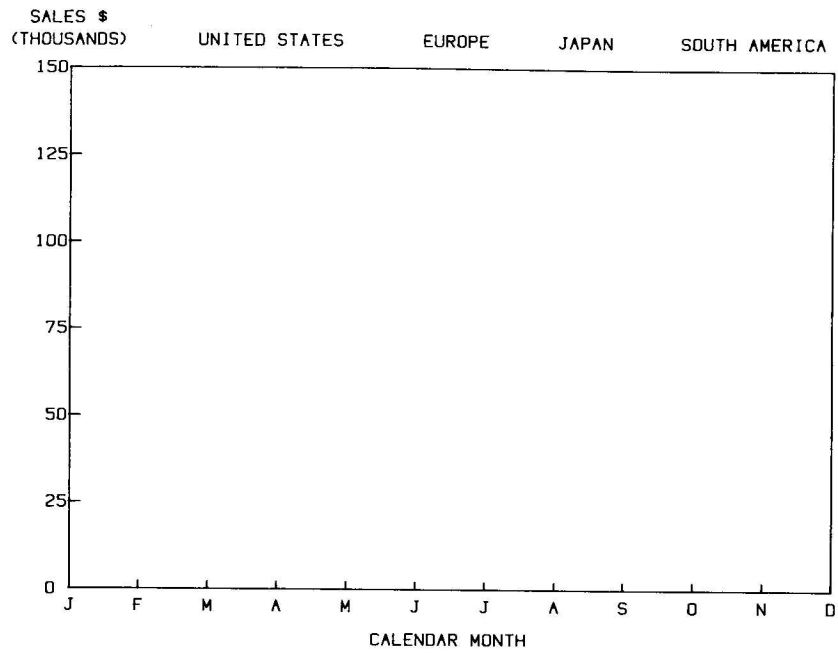
The Y-axis is created in a similar manner, except the loop's index is used for the label value and two different CP commands are used for labels of three digits and labels of less than three digits. The Y-axis title is centered above the axis.

Following the axis routine is the command which labels the regions for the legend. It is drawn now while the label size is small and the narrow pen is installed. Note that the label statements contain the spaces necessary to space the legend across the top of the graph. These lines were inserted near the end of the creation process and involved trial and error to achieve satisfactory results. The lines for the legend will be drawn later as each line of data is plotted.

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 "PA 1,",Y,"YT;"
140 IF Y<100 THEN PRINT "CP-3,-.25;LB";Y;"ETX"
150 IF Y>99 THEN PRINT "CP-4,-.25; LB";Y;"ETX"
160 NEXT Y
170 PRINT "PA1,150 CP-3.5,2 LBSALES $ETXCP-9,-1"
180 PRINT "LB(THOUSANDS)       UNITED STATES       ETX"
190 PRINT "LBEUROPE       JAPAN       SOUTH AMERICAETX"
```

Here's what the graph looks like so far.



Adding Color and Emphasis

Because the most important part of the graph is the data and title, we will emphasize these using wide pens in one of two colors. (Wide pens may be purchased from Hewlett-Packard or your dealer; part numbers are listed in the Operator's Manual under Accessories Available.) This program pauses and displays a message on the CRT as a reminder to change pens. The technique you use will depend on your computer system. This program also removes the prompt as the first step when the program continues. You may want to use only two pens in your whole plot. If so, you can use two colors of the same width or one wide and one narrow pen and run your program from beginning to end. If you are not going to change pens, either delete the PAUSE statements or continue your program immediately when the prompt appears on your CRT.

A word to the wise: whenever you do want to change pens, insert a pause in your program. It ensures you will make that pen change at the proper time and that the pen will not hit your hand as you try to change pens while plotting is in progress. If a pen was in the pen holder when your program paused, store it in its stall, manually remove the old pen from the stall, and replace it with the new pen. Then select the new pen, using front-panel controls, before you restart your program. If you do not reload the pen holder, your program will continue

plotting without a pen until it encounters an SP command. You can lessen the manual intervention by storing the pen using SP0 as the last HP-GL instruction before any pause, and by issuing a pen select command as the first HP-GL instruction after the pause.

Program lines to pause, change pens, and title the graph using a wide pen follow. Remember when you run the complete program to remove the old pens and load wide pens directly into the left and right stalls when the message appears. The SP1 command here, the first command after the program pause, assures that the pen holder is loaded so all subsequent lines will be drawn.

```
200 PRINT "SP0;"
210 DISP "CHANGE TO WIDE PENS"
220 PAUSE
230 DISP " "
240 PRINT "SP1 PA6,150 SI.4,.6 CP-9.5,2.0"
250 PRINT "LB1981 SALES BY REGION%"
```

Plotting Your Data

We are now ready to draw lines. Each of the four data lines on this graph is drawn using a different technique. The first two lines are drawn by plot commands with parameters included when the program was written. Hence, if the data changes, it will be necessary to change the plot commands in the program.

The first line (the bottom-most line on the graph) is drawn with pen 1 using a dashed line type. The program takes full advantage of the plotter's relatively free syntax and uses spaces to delineate parameters. Send the character strings to the plotter exactly as shown. Be sure to enter those spaces; if the spaces are removed, the plotter will try to plot one very large number and you won't plot the line.

After drawing the line, the pen moves to the legend area below the graph title and draws a short line. The PU command causes the line type pattern to begin again at the beginning of this line.

The second line is also plotted using plot commands with fixed parameters. These plot commands use the stricter syntax of the 9872 or 7225 plotters and would be accepted by any HP plotter programmed in HP-GL. The line type used consists of long and short dashes; the line is drawn with pen 2. After the data 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:

```
260 PRINT "SP1;LT3,6;PA1 23PD2 25 3 18 4 22 5 23"
270 PRINT "PD6 27 7 27 8 25 9 24 10 28 11 27 12 27PU"
280 PRINT "PA7.8,165 PD9.3,165 PU"
290 PRINT "SP2;LT6,8;PA1,45;PD;PA2,50,3,52,4,53,5,52"
300 PRINT "PD6,51,7,55,8,56,9,56,10,58,11,58,12,60PU"
310 PRINT "PA10.1,165 PD11.6,165 PU"
```

The third line is plotted from data read by the program at execution time using a FOR...NEXT loop and a READ statement. This technique would be used to plot a graph 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 command with no parameters. Since we are using variables as plot parameters, you need to be sure they are sent to the plotter with a space between numeric variables. Computers often send a leading and/or trailing blank or allow for a sign space before numeric variables. The 7470 will treat a blank, comma, or sign as a separator between numeric parameters. Know your computer before sending variables with plot commands. As with the two previously drawn lines, after the line is plotted, the corresponding line is placed in the legend.

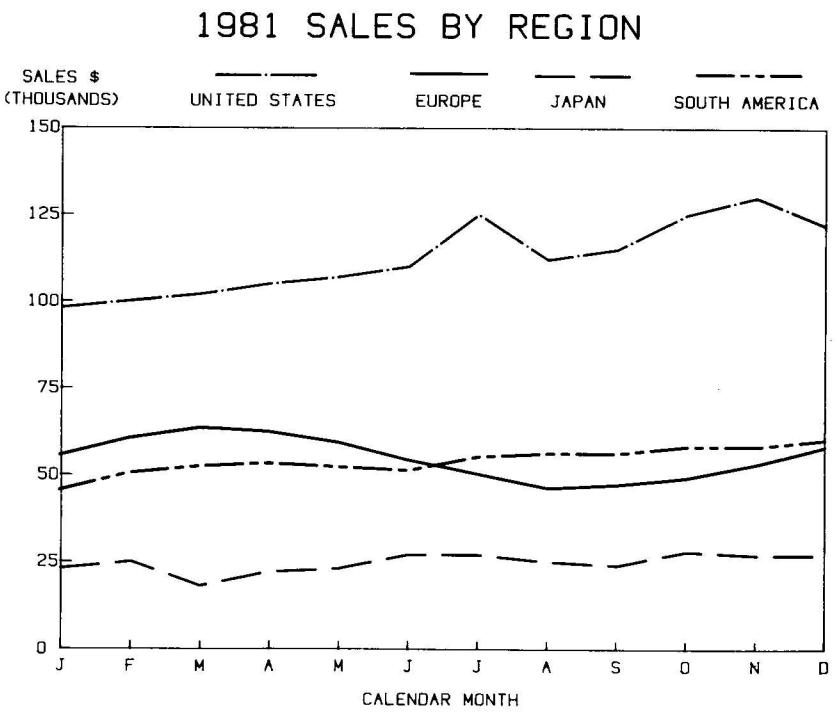
The loop to plot this third line and the statements to place a line in the legend are:

```
320 PRINT "LT"
330 FOR X=1 TO 12
340 READ Y
350 PRINT "PA";X;Y;"PD"
360 NEXT X
370 PRINT "PU6,165PD7.1,165PU"
410 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 statements, one with the pen up and the other with the pen down. In this program, a zero as a 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 legend line is drawn at the end of the subroutine, completing the graph.

The program lines to change pens and line type, and the subroutine itself are listed here, followed by a reduced version of the completed plot.

```
380 PRINT "SP1;LT4,6"
390 GOSUB 1000
1000 ! PLOTTING SUBROUTINE
1010 READ X,Y,P
1020 IF P=1 THEN PRINT "PD";X;Y
1030 IF P=0 THEN PRINT "PU";X;Y
1040 IF P=3 THEN 1090
1050 DATA 1,98,0,2,100,1,3,102,1,4,105,1,5,107,1,6,110,1
1060 DATA 7,125,1,8,112,1,9,115,1,10,125,1,11,130,1
1070 DATA 12,122,1,0,0,3
1080 GOTO 1010
1090 PRINT "LT4,6 PU3.2,165 PD4.7,165SP0;"
1100 RETURN
```



Listing

A complete listing of the program follows. This listing contains all the BASIC statements necessary to have this program run on an HP-85 computer with an HP-IB interface and the plotter set to address 5. When the plotter is used with an RS-232-C interface, line 10 should be replaced by other lines which send the escape code sequences necessary to turn on the plotter and establish handshaking. In some PRINT statements, semicolons or commas are used to ensure that HP-GL commands will have the necessary separators or no extra spaces. You may need to make changes for your computer's BASIC, or you can use some other programming language and send the strings of HP-GL commands using your language's output statements and looping techniques.

NOTE: The end-of-text character ↖ is equivalent to $\overline{N}$ on the HP-85's display and internal printer. ($\overline{N}$ is obtained on the HP-85 by pressing **CTRL** and **C** simultaneously. On many computers, you can also use the CHR$(3) function to generate the end-of-text character.) This program listing was produced on an HP 7310 printer. ■

```
10 PRINTER IS 705,80
20 PRINT "IN;SP1;IP1250,750,9250,6250;"
30 PRINT "SC1,12,0,150;"
40 PRINT "PU1,0 PD 12,0,12,150,1,150,1,0 PU"
50 PRINT "SI.2,.3;TL1.5,0"
60 FOR X=1 TO 12
70 PRINT "PA";X,",0; XT;"
80 READ A$
90 PRINT "CP-.33,-1;LB";A$;"↖"
100 NEXT X
110 PRINT "PA6.5,0;CP-7,-2.5; LBCALENDAR MONTH↖"
120 FOR Y=0 TO 150 STEP 25
130 PRINT "PA 1,",Y,"YT;"
140 IF Y<100 THEN PRINT "CP-3,-.25;LB";Y;"↖"
150 IF Y>99 THEN PRINT "CP-4,-.25; LB";Y;"↖"
160 NEXT Y
170 PRINT "PA1,150 CP-3.5,2 LBSALES $↖CP-9,-1"
180 PRINT "LB(THOUSANDS)      UNITED STATES      ↖"
190 PRINT "LBEUROPE      JAPAN      SOUTH AMERICA↖"
200 PRINT "SP0;"
210 DISP "CHANGE TO WIDE PENS"
220 PAUSE
230 DISP " "
240 PRINT "SP1 PA6,150 SI.4,.6 CP-9.5,2.0"
250 PRINT "LB1981 SALES BY REGION↖"
260 PRINT "SP1;LT3,6;PA1 23PD2 25 3 18 4 22 5 23"
270 PRINT "PD6 27 7 27 8 25 9 24 10 28 11 27 12 27PU"
280 PRINT "PA7.8,165 PD9.3,165 PU"
290 PRINT "SP2;LT6,8;PA1,45;PD;PA2,50,3,52,4,53,5,52"
300 PRINT "PD6,51,7,55,8,56,9,56,10,58,11,58,12,60PU"
310 PRINT "PA10.1,165 PD11.6,165 PU"
```

(Program listing continued)

```
320 PRINT "LT"
330 FOR X=1 TO 12
340 READ Y
350 PRINT "PA";X;Y;"PD"
360 NEXT X
370 PRINT "PU6,165PD7.1,165PU"
380 PRINT "SP1;LT4,6"
390 GOSUB 1000
400 DATA "J","F","M","A","M","J","J","A","S","O","N","D"
410 DATA 55,60,63,62,59,54,50,46,47,49,53,58
420 STOP
1000 ! PLOTTING SUBROUTINE
1010 READ X,Y,P
1020 IF P=1 THEN PRINT "PD";X;Y
1030 IF P=0 THEN PRINT "PU";X;Y
1040 IF P=3 THEN 1090
1050 DATA 1,98,0,2,100,1,3,102,1,4,105,1,5,107,1,6,110,1
1060 DATA 7,125,1,8,112,1,9,115,1,10,125,1,11,130,1
1070 DATA 12,122,1,0,0,3
1080 GOTO 1010
1090 PRINT "LT4,6 PU3.2,165 PD4.7,165SP0;"
1100 RETURN
1110 END
```

Advanced Programming Tips

Filling and Hatching

Two kinds of area fill are commonly used in bar graphs 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 lines in two directions at 90 degree angles, we call the hatching crosshatching. Sometimes a graph will have both narrow and wide hatching or crosshatching, the wide hatching having more space between the lines than the narrow.

Filling a Bar

The following two program segments, together with lines 10 to 100 and 400 of this chapter's program, will each fill a bar which represents the March data for line 1, i.e., 3,18 (see line 260, in the program). To create an aesthetically pleasing and easily comprehendible bar graph, the bar is centered over the X data point and is slightly wider than one-half the distance between data points on the X-axis. The increment variable P depends on pen width. A value of P = 20 plotter units is suitable for a wide pen and 10 for a narrow pen.

The first program segment should be used when plotting on paper. Notice the pen does not lift; the routine is faster and prolongs pen-tip life by limiting up/down moves. The second segment should be used when plotting on transparency film to achieve uniform ink distribution.

The first routine performs the following tasks:

1. Obtains, in plotter units, the coordinates of the corners of the bar.
2. Turns scaling off so plotting is in plotter units. This routine can, therefore, be used in any program, and there is no need to recompute the increment P for different scaling in different graphs.
3. Beginning at the X,Y_{min} value, draws a line to the top of the bar, moves over slightly less than one pen width, and draws to the bottom of the bar.
4. Increments the X-value one pen width and repeats step 3 until the bar is filled.
5. Rescales the plot to the original scaling.

The second routine repeatedly moves with the pen up to the X-coordinate at the base of the bar and draws a line to the top of the bar. All fill lines are drawn in the same direction.

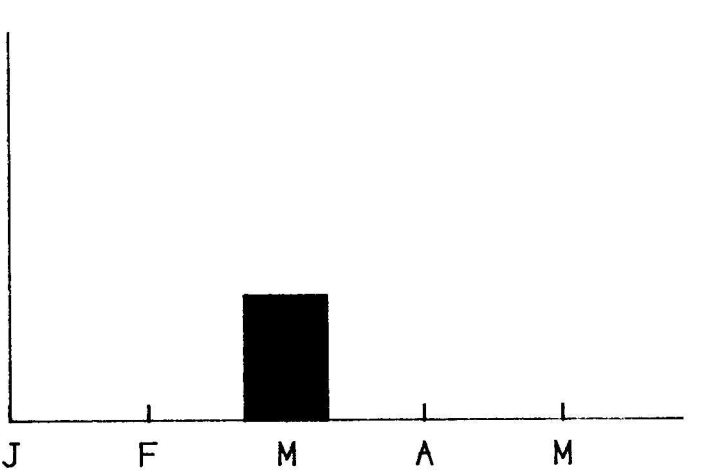
Segment 1 — Plotting on Paper

```
120 PRINT "PA2.7,0,PD,2.7,18,3.3,18,3.3,0,2.7,0;PU;"
130 PRINT "OA;"
140 ENTER 705 ; A,B,C
150 PRINT "PA2.7,18;OA;"
160 ENTER 705 ; D,E,F
170 PRINT "PA3.3,18;OA;"
180 ENTER 705 ; G,H,I
190 PRINT "PA2.7,0;SC;"
200 P=20
210 FOR X=A TO G-P STEP 2*P
220 PRINT "PD";X;B;X;E
230 PRINT "PD";X+P;E;X+P;B
240 NEXT X
250 PRINT "PU;SC1,12,0,150;"
```

Segment 2 — Plotting on Transparency Film

```
120 PRINT "PA2.7,0,PD,2.7,18,3.3,18,3.3,0,2.7,0;PU;"
130 PRINT "OA;"
140 ENTER 705 ; A,B,C
150 PRINT "PA2.7,18;OA;"
160 ENTER 705 ; D,E,F
170 PRINT "PA3.3,18;OA;"
180 ENTER 705 ; G,H,I
190 PRINT "PA2.7,0;SC;"
200 P=20
210 FOR X=A TO G STEP P
220 PRINT "PU";X;B;"PD";X;E
230 NEXT X
240 PRINT "PU;SC1,12,0,150;"
```

Plot showing filled bar



Hatching a Bar

The following program segment, together with lines 10 to 100 and 400 of this chapter's program, hatches a bar which represents February data for line 2, i.e., 2,50. The XT instruction was deleted from line 70 to omit drawing the X-ticks. Again the bar is centered over the X data point. In this segment, the increment variable P is the distance between hatching lines and determines whether a wide or narrow hatch pattern is drawn. You may want to make further refinements depending on pen width and bar width and height. The bars are shown here actual size with P set at 100 and 300. The locations of the variables are shown on the first bar and should help you understand the program listing.

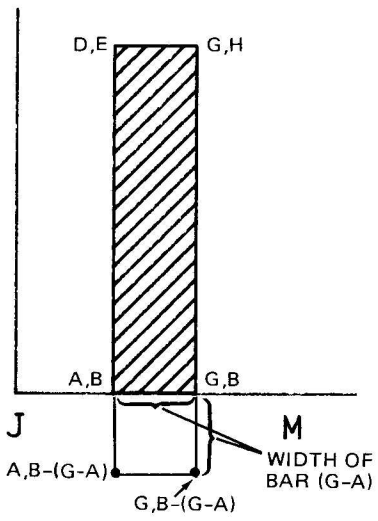
For plots on transparency film or to make hatch lines more uniform, you should slow the pen velocity using the VS instruction.

The routine performs the following tasks.

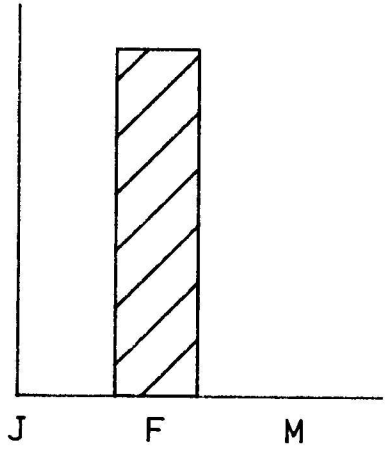
1. and 2. (Same as solid fill algorithm.)
3. Using the output obtained in step 1, sets the window to be the bar we wish to hatch.
4. Beginning the width of the bar below the Y_{min} value, plots a line at a 45 degree angle to the opposite side of the bar, increments the Y-value and continues the process until the top of the bar is reached.
5. Resets the scaling and window to their previous values.

To crosshatch a bar, add program lines to draw from the left to the right side of the bar, starting at G, B-(G-A).

```
110  PRINT "PA1.7,0,PD,1.7,50,2.3,50,2.3,0,1.7,0;PU;"
120  PRINT "OA;"
130  ENTER 705 ; A,B,C
140  PRINT "PA1.7,50;OA;"
150  ENTER 705 ; D,E,F
160  PRINT "PA2.3,50;OA;"
170  ENTER 705 ; G,H,I
180  PRINT "PA1.7,0;SC;"
190  P=100
200  PRINT "IW";A;B;G;H
210  FOR Y=B-(G-A) TO E STEP P
220  PRINT "PU";A;Y;"PD";G;Y+(G-A)
230  NEXT Y
240  PRINT "PU;SC1,12,0,150;IW;"
```



P = 100



P = 300

Filling Segments of Pie Charts

The algorithms to fill slices of pie charts are much more complex because the areas are not rectangular. Software packages such as the Graphics Presentations Pacs for various HP desktop computers make it easy to draw pie charts with area fill. You may wish to purchase such software so you do not have to invest hours of programming time in order to create filled pie charts.

Notes

Chapter 9
HP-IB Interfacing

What You'll Learn in This Chapter

This chapter is only for 7470 owners with an HP-IB interface. HP 7470s with Option 002 have an HP-IB interface.

In this chapter you'll learn how to operate your plotter when it is connected to a computer using the Hewlett-Packard Interface Bus (HP-IB), which conforms to ANSI/IEEE 488-1978 specifications. This chapter defines the 7470's implementation of the bus. Also included are addressing the 7470, the listen-only mode, reaction to bus clear commands, serial and parallel polling, addressing the 7470 as a talker or listener, and examples of sending and receiving data using a variety of computers.

This chapter assumes you have a working knowledge of the HP-IB; however, if you wish to refresh your memory on HP-IB structure, refer to Appendix A of this manual, entitled An HP-IB Overview.

HP-IB Implementation on the 7470

The HP-IB conforms to ANSI/IEEE 488-1978 specifications, and direct interconnection of the HP-IB is via a connector on the rear panel.

The HP-IB functions implemented in the 7470 are as follows:

1. Source Handshake (SH1)
2. Acceptor Handshake (AH1)
3. Talker (T2)
4. Listener (L2)
5. Service Request (SR1)
6. No Remote Local (RL0)
7. Parallel Poll (PP0 if lon ; PP2 if addr <8 ; PP0 otherwise)
8 Device Clear (DC1)
9. No Device Trigger (DT0)
10. No Controller (C0)

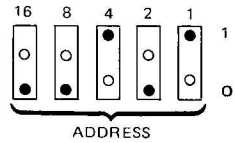
Interface Switches and Controls

The 7470 plotter functions in either of two modes, addressable mode and listen-only mode. In addressable mode, the plotter can function as a talker or as a listener depending on the instructions it receives from the controller. In listen-only mode, it can only listen and it hears all activity on the bus.

Addressing the Plotter

Rear panel switches provide for selection of the plotter address or listen-only mode. Each HP-IB interface can have as many as 15 devices connected to it, set to different specific address codes. The plotter can be set to any one of 31 HP-IB addresses, ranging from 0 through 30. Each address can be selected by setting the switches on the rear panel to the appropriate binary bit positions for the particular address value desired. The address selected establishes the 7470's device address. When using the plotter with an HP desktop computer, do not use 21 which is reserved for the desktop computer's address. When not using an HP desktop computer, be sure the computer and plotter do not have the same address. (Refer to the documentation for your computer.) Address 31 is used to set the plotter to listen-only mode.

The plotter is set to an address code of 05 at the factory. This corresponds to a listen character of % and a talk character of E. Check the following figure for the factory-set address switch positions.



The following table lists the address switch positions for each address value.

Address Characters		Address Switch Settings					Address Codes		
Listen	**Talk**	**16**	**8**	**4**	**2**	**1**	**Decimal**	**Octal**	
SP	@	0	0	0	0	0	0	0	
!	A	0	0	0	0	1	1	1	
"	B	0	0	0	1	0	2	2	
#	C	0	0	0	1	1	3	3	
$	D	0	0	1	0	0	4	4	
%	E	0	0	1	0	1	5	5	← preset
&	F	0	0	1	1	0	6	6	
'	G	0	0	1	1	1	7	7	
(	H	0	1	0	0	0	8	10	
)	I	0	1	0	0	1	9	11	
*	J	0	1	0	1	0	10	12	
+	K	0	1	0	1	1	11	13	
,	L	0	1	1	0	0	12	14	
−	M	0	1	1	0	1	13	15	
.	N	0	1	1	1	0	14	16	
/	O	0	1	1	1	1	15	17	
0	P	1	0	0	0	0	16	20	
1	Q	1	0	0	0	1	17	21	
2	R	1	0	0	1	0	18	22	
3	S	1	0	0	1	1	19	23	
4	T	1	0	1	0	0	20	24	
5	U	1	0	1	0	1	21	25	← Reserved for HP Desktop Computer Address
6	V	1	0	1	1	0	22	26	
7	W	1	0	1	1	1	23	27	
8	X	1	1	0	0	0	24	30	
9	Y	1	1	0	0	1	25	31	
:	Z	1	1	0	1	0	26	32	
;	[	1	1	0	1	1	27	33	
<	\	1	1	1	0	0	28	34	
=	]	1	1	1	0	1	29	35	
>	^	1	1	1	1	0	30	36	
?	_	1	1	1	1	1	31	37	← Sets Listen-only Mode

Bus Commands

Reaction to Bus Commands DCL, SDC, and IFC

The computer can set all devices on the HP-IB system to a predefined or initialized state by sending the device clear command, DCL. The computer can also set selected devices to a predefined or initialized state by sending a selected device clear command, SDC, along with the addresses of the devices. The basic difference is that devices will obey SDC only if they are addressed to listen, whereas DCL clears all devices on the bus. The interface clear command, IFC, is used by the computer to override all bus operations and return the bus to a known quiescent state.

Upon receipt of either a DCL, SDC, or IFC command, the plotter resets the I/O to begin accepting a new instruction, and disables any current output. Any partially parsed HP-GL instruction or parameters will be lost.

The device clear and interface clear commands *do not* reset parameters in the plotter to their default values. They are not the same as the HP-GL commands, DF or IN.

Serial and Parallel Polling

Polling is the process used by the computer to determine which device on the HP-IB bus has initiated a require service message. The conditions which will cause the require service message to be sent to the computer are defined by the input mask instruction, IM, in Chapter 1.

The Serial Poll

A serial poll enables the computer to learn the status or condition of devices on the bus. It is commonly used by the computer to determine who is requiring service.

The serial poll is so named because the computer polls devices one at a 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 (Chapter 7). The S-mask parameter of the input mask instruction, IM, is used to specify which status byte conditions will send the service request message and when polled, respond with request service. Unless the user changes the S-mask value from the default setting of 0, the plotter will never give a positive response to a serial poll, i.e., request service (see The Input Mask Instruction, IM, Chapter 1). Bit position 6 of the status byte will be set to 1 (if the S-mask value is not 0) when any of the conditions designated by the S-mask are true. Bit position 6 will be set to 0 after all conditions which would cause a service request no longer exist. See IM, Chapter 1, and OS, Chapter 7. Until bit position 6 has been reset to 0, no additional service request messages, and therefore, no responses to a serial poll are possible.

A computer must issue special commands to initiate and terminate a serial poll. During a serial poll, a device must be instructed to talk and the computer to listen. Therefore, a serial poll cannot be executed when a plotter is in listen-only mode.

The Parallel Poll

Parallel polling can only be done to plotters with an address 0 through 7. Plotters with address settings from 8 through 30 cannot respond to a parallel poll. 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 logical 1 response to a parallel poll. The response to a parallel poll is limited to setting the appropriate data line to a logical 1. The line used is determined by the plotter's address value as shown in the table below:

Plotter Address	Parallel Poll Bit Position	HP-IB Data Line Number	
0	7	8	
1	6	7	
2	5	6	
3	4	5	
4	3	4	
5	2	3	Plotter Preset Address
6	1	2	
7	0	1	

To execute a parallel poll, the controller sets the ATN and EOI lines to 1. The controller reads the eight data lines, and determines from these lines which instrument on the bus is requesting service. The computer then sends the parallel poll disable command. Not all computers have parallel poll capability.

It is important to remember that the 7470 will not send a logical 1 unless the P-mask bit 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.

Positive responses to parallel polls will continue to occur until all bits of the status byte included in the P-mask value have been reset to 0. (See The Output Status Instruction, OS, Chapter 7.)

Addressing the 7470 as a Talker or Listener

In order to communicate effectively with the 7470 plotter, it is important that you completely understand the addressing protocol of your computer. Therefore, you may wish to review this aspect of your computer before proceeding.

Computers with No High Level I/O Statements

On low level computers, addressing devices on the HP-IB bus is accomplished using mnemonics, such as CMD, which serve as the "bus command."

When bus commands are necessary, a typical addressing sequence is

<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 unlisten command ("?") plays a vital role in this sequence. It is important that a device receive only the data that is intended for it.

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.

Computers with High Level I/O Statements

In more powerful computers, higher level input/output (I/O) statements are used to specify device addresses on the HP-IB bus. In these cases, the addressing protocol (unlisten, talk, listen) is a function of the computer's internal operating system and need not be of concern to the user.

Sending and Receiving Data

Computer-to-Plotter

Transmitting data from a computer to the plotter is typically accomplished using I/O statements such as WRITE, PRINT, PRINT#, or OUTPUT. The following examples of sending program data to the plotter from various computers are only intended to illustrate the necessity for understanding the I/O statement protocol implemented by your computer. Each of these examples will cause the plotter to label the identity of the computer sending data, beginning at the X,Y coordinates 1000,2000. The examples involve sending both character string and numeric data as variables, and constants or literals.

HP 9825 and 9826 HPL Example:

```
0: fxd 0;dim A$[13]
1: " SENDING DATA"→A$
2: 2000→Y
3: 9826→B
4: wrt 705,"SP1;PA1000,",Y
5: wtb 705,"LBHP",str(B),A$,3
6: end
```

A terminator is sent by the 9825/9826 at the end of a wrt statement.

Result: HP 9826 SENDING DATA

9826 BASIC Example:

```
10    PRINTER IS 705
20    A$=" SENDING DATA"
30    B=9826
40    Y=2000
50    PRINT "SP1;PA1000,",Y
60    PRINT USING "K";"LBHP ",B,A$,"%"
70    END
```

A terminator is sent by the 9826 at the end of a PRINT statement.

Result: HP 9826 SENDING DATA

HP 9835/9845 Example:

```
10   PRINTER IS 7,5
20   A$=" SENDING DATA"
30   B=9835
40   C=9845
50   Y=2000
60   PRINT "SP1;PA1000,";Y
70   PRINT USING "K";"LBHP ",B,"/",C,A$,CHR$(3)
80   END
```

A terminator is sent by the computer at the end of a PRINT statement.

Result: HP 9835/9845 SENDING DATA

HP 2647 Example:

```
10 ASSIGN "H#5" TO #1
20 DIM A$[13]
30 A$="SENDING DATA"
40 B=2647
50 Y=2000
60 PRINT #1;"SP1;PA1000,",Y
70 PRINT #1;"LBHP",B,A$,CHR$(3)
80 END
```

A terminator is sent by the 2647 at the end of PRINT #1 statements.

Result: HP 2647 SENDING DATA

HP-83/85 Example:

```
10 PRINTER IS 705
20 A$="SENDING DATA"
30 B=85
40 Y=2000
50 PRINT "SP1;PA1000,",Y
60 PRINT "LBHP";B;A$;"ᴺ"
70 END
```

A terminator is sent by the computer following PRINT statements.

Result: HP 85 SENDING DATA

TEK 4051 Example:

```
100 DIM A$[13],B$[1]
110 A$=" SENDING DATA
120 Y=2000
130 B=4051
135 B$=CHR(3)
140 PRINT @5:"SP1;PA1000,";Y;";"
150 PRINT @5:"LBTEK";B;A$;B$
160 END
```

No terminator is sent by the TEK 4051. It must, therefore, be included in each PRINT @ 5 statement if the last HP-GL command in the line requires one. In line 140, all characters after the Y may be omitted, since the terminator is optional with the PA command.

Result: TEK 4051 SENDING DATA

Commodore PET* 2001 and 8032 Example:

```
10 OPEN 5,5
20 DIM A$(13)
30 A$=" SENDING DATA"
40 B=2001
50 Y=2000
60 PRINT#5,"SP1;PA1000,";STR$(Y)
70 PRINT#5,"LBPET ";B;A$;CHR$(3)
80 END
```

A terminator is sent by PET at the end of the PRINT #5 statement.

Result: PET 2001 SENDING DATA

Apple* II Applesoft BASIC Example:

```
10  PR# 3: IN# 3
20 Z$= "WT%" + CHR$ (26)
30  DIM A$(12)
40 A$= " SENDING DATA"
50 Y= 2000
60  PRINT Z$; "SP1;PA1000,",Y
70  PRINT Z$; "LBAPPLE II ";A$;CHR$ (3)
80  PR# 0: IN# 0
90  END
```

*Commodore PET is a trademark of Commodore Business Machines, Inc.
Apple is a trademark of Apple Computer, Inc.

Result: APPLE II SENDING DATA

The PR# 3: IN# 3 statement must be included in each program before instructions can be sent to the plotter. These statements assume the IEEE-488 interface card (HP-IB) is in slot three of the computer. The string Z$ addresses the plotter at address 5 to listen. It must be included in every print statement which sends HP-GL commands to the plotter. The PR# 0: IN# 0 statement directs keyboard output to the display and must be included before the end of the program or before anything can be printed on the display.

Plotter-to-Computer

Typically, the computer obtains output information from the plotter by using I/O statements such as READ, INPUT, or ENTER. Sometimes these statements are available only in I/O ROMs, such as in the HP Series 80 computers. Check your computer documentation or ask your HP salesperson to determine if your system requires a special I/O ROM. The following examples of obtaining output data from the plotter using various computers are only intended to illustrate the necessity for understanding the I/O statement protocol implemented on your computer. Each of these examples commands the pen to move to plotter coordinates X = 1000, Y = 1000 and then output the current pen position and the plotter identifier string to the computer.

HP 9825 and 9826 HPL Example:

```
0: fxd 0;dim A$[5]
1: wrt 705,"PA1000,1000;OC"
2: red 705,A,B,C
3: wrt 705,"OI"
4: red 705,A$
5: dsp A,B,C,A$
6: end
```

Displayed current pen position and identification.

```
1000  1000  0 7470A
```

HP 9826 BASIC Example:

```
10    PRINTER IS 705
20    PRINT "PA1000,1000;OC"
30    ENTER 705;A,B,C
40    PRINT "OI"
50    ENTER 705;A$
60    DISP A,B,C,A$
70    END
```

Displayed current pen position and identification.

```
1000        1000        0        7470A
```

HP 9835/9845 Example:

```
10   PRINTER IS 7,5
20   PRINT "PA1000,1000;OC"
30   ENTER 705;A,B,C
40   PRINT "OI"
50   ENTER 705;A$
60   DISP A,B,C,A$
70   END
```

Displayed current pen position and identification.

```
1000          1000          0          7470A
```

HP 2647 Example:

```
10 ASSIGN "H#5" TO #1
20 PRINT #1;"PA1000,1000;OC"
30 READ #1;A,B,C
40 PRINT #1;"OI"
50 READ #1;A$
60 PRINT A,B,C,A$
70 END
```

Displayed current pen position and identification.

```
1000          1000          0          7470A
```

HP-85/86/87 Example:*

```
10 PRINTER IS 705
20 PRINT "PA1000,1000;OC"
30 ENTER 705 ; A,B,C
40 PRINT "OI;"
50 ENTER 705 ; A$
60 DISP A,B,C,A$
70 END
```

Displayed current pen position and identification.

```
1000          1000
0             7470A
```

*Requires I/O ROM HP Part Number 00087-15003.

TEK 4051 Example:

```
100 DIM A$[5]
110 PRINT @5:"PA1000,1000;OC;"
120 INPUT @5:A,B,C
130 PRINT @5:"OI;"
140 INPUT @5:A$
150 PRINT A,B,C,A$
160 END
```

Displayed current pen position and identification.

1000	1000	0	7470A

Commodore PET 2001 Example:

```
10 OPEN 5,5
20 PRINT#5,"PA1000,1000;OC"
30 INPUT#5,A,B,C
40 PRINT#5,"OI"
50 INPUT#5,A$
60 PRINT A,B,C,A$
70 END
```

Displayed current pen position and identification.

1000	1000	0	7470A

Commodore PET 8032 Example:

On the PET 8032, all alphabetic characters are displayed as lowercase. This is true for both BASIC program statements and for the plotter's response.

A dummy string variable should be included at the end of every input command which reads data from the plotter because the PET 8032 sends an untalk command after it receives a carriage return character. Since the plotter with an HP-IB interface terminates all output with a carriage return followed by a line feed, the line feed must be read into this dummy string variable in order to clear the plotter's output buffer for future output.

```
10 OPEN 5,5
20 PRINT#5,"PA1000,1000;OC"
30 INPUT#5,A,B,C,B$
40 PRINT#5,"OI"
50 INPUT#5,A$,B$
60 PRINT A,B,C,A$
70 END
```

Displayed current pen position and identification.

```
1000        1000        0          7470a
```

Apple II Applesoft BASIC Example:

```
10  PR# 3: IN# 3
20 Z$= "WT%" + CHR$ (26)
30 Y$= "RDE" + CHR$ (26)
40  PRINT Z$; "PA1000,1000;OC;"
50  PRINT Y$;
60  INPUT A,B,C
70  PRINT Y$;
80  INPUT D$
90  PRINT Z$; "OI"
100  PRINT Y$;
110  INPUT A$
120  PRINT Y$
130  INPUT D$
140  PR# 0: IN# 0
150  PRINT A,B,C,A$
160  END
```

Displayed current pen position and identification.

```
1000              1000              0
7470A
```

For an explanation of PR# 3, Z$ and PR# 0, refer to the Apple II example in the prior section. The string Y$ instructs the plotter at address 5 to talk. The Apple II sends an untalk command after it receives a carriage return character. The plotter with an HP-IB interface terminates all output with a carriage return followed by a line feed. Therefore, in order to clear the plotter's buffer for future output, another talk instruction and another input statement containing a dummy variable (D$ in this program) must follow the input statement which reads parameters of the plotter output statement. The additional talk and input instructions will read the line feed character, thus clearing the plotter's buffer.

Notes

Chapter 10
RS-232-C/CCITT V.24 Interfacing

What You'll Learn in This Chapter

This chapter is only for 7470 owners with an RS-232-C interface. HP 7470s with Option 001 have an RS-232-C interface.

This chapter describes how to connect the plotter, terminal, and computer in a modem or hardwire environment. It also discusses connecting the interface, pin allocations in the connector, baud rates, stop bits, and transmission errors. A tutorial description of the four handshaking methods, hardwire handshake, Xon-Xoff handshake, enquire/acknowledge handshake, and software checking handshake, is included. The last part of the chapter is devoted to the 14 device control instructions. The syntax of device control instructions is given, followed by a detailed section on each instruction. It is important to be able to use these instructions properly to establish communications with the plotter in your operating environment. You need to master the material in this chapter so you can successfully send HP-GL commands to the plotter.

NOTE: All information in this chapter applies equally to RS-232-C and CCITT V.24 interfaces. For purposes of simplicity, both are referred to as RS-232-C. ■

Setting Up Your RS-232-C Plotter: a Checklist

The following steps should be followed when interfacing the 7470 plotter with a computer using an RS-232-C interface.

1. Determine which installation and operating environment, described in the first few pages of this chapter, matches your system.
2. Check that you have the required cables and connect the plotter as pictured in the section which describes the environment chosen in step 1. Information necessary when constructing your own cable is found in the section Connecting the RS-232-C Interface.
3. Determine if parity checking is used on your system and set the rear panel parity switches **S1** and **S2** accordingly. Refer to the 7470 Operator's Manual.
4. Determine the baud rate at which your computer sends data and set the rear panel switches **B1** through **B4** accordingly. Refer to the 7470 Operator's Manual.
5. Determine which handshake your system uses. The four kinds of handshakes are described in the section entitled Handshaking. Note which device control instructions are used to establish that handshake. Since handshaking is often a function of your operating system, you may need to refer to the manuals for your computer to determine which parameters you must set and to what values.
6. In the last part of this chapter, read about the instructions you will use to set up the handshake you have chosen.

Plotter Environments

There are three possible ways to position the 7470 plotter in a computer system. They are described in the following pages; you need only read the section which applies to your system.

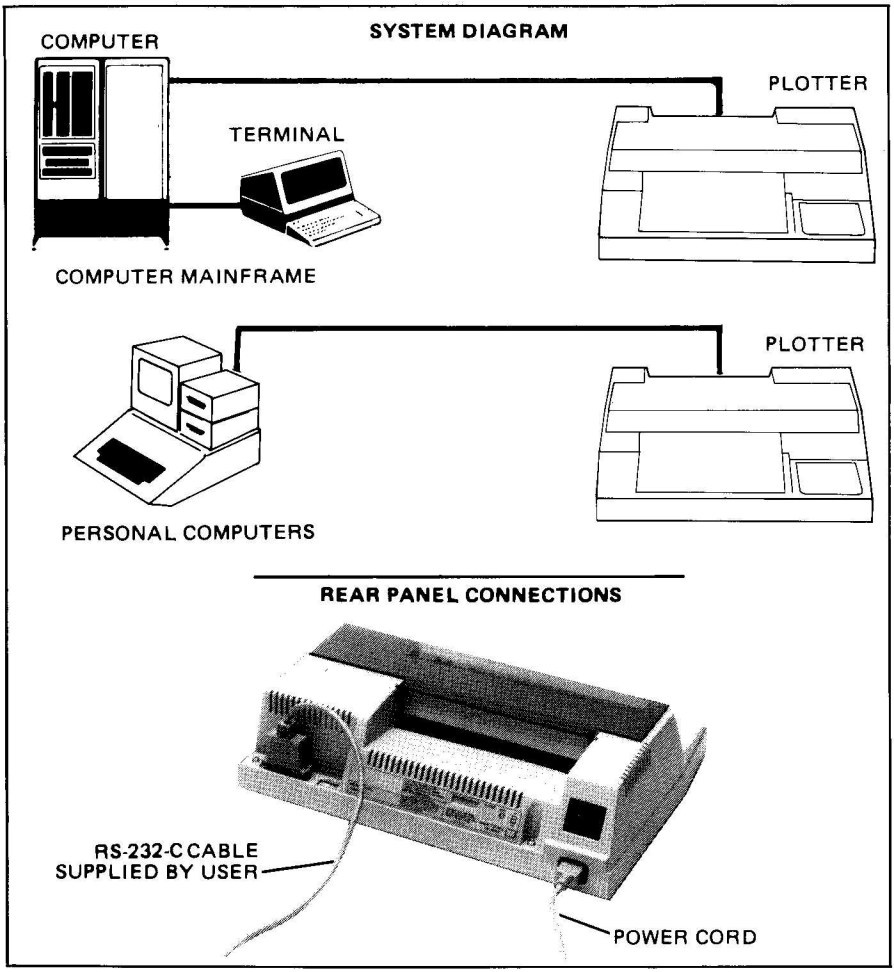
Once the plotter has been connected in a system, it can be placed in an operating state. The operating states which can be accessed in a given environment are described in the operation section for each of the three environments.

Using a Plotter Directly Connected to a Computer Mainframe or Personal Computer

Installation

In this type of system, the plotter is connected directly to a computer and is usually adjacent to it. Entry to the computer is by a keyboard or

terminal through a separate port, rather than through the plotter. This is sometimes referred to as an endline or stand-alone environment. Diagrams of this type of system for both large and personal computers are shown below, along with a picture of the rear panel connection.



Plotter Connection with a Computer Mainframe or Personal Computer

Operation

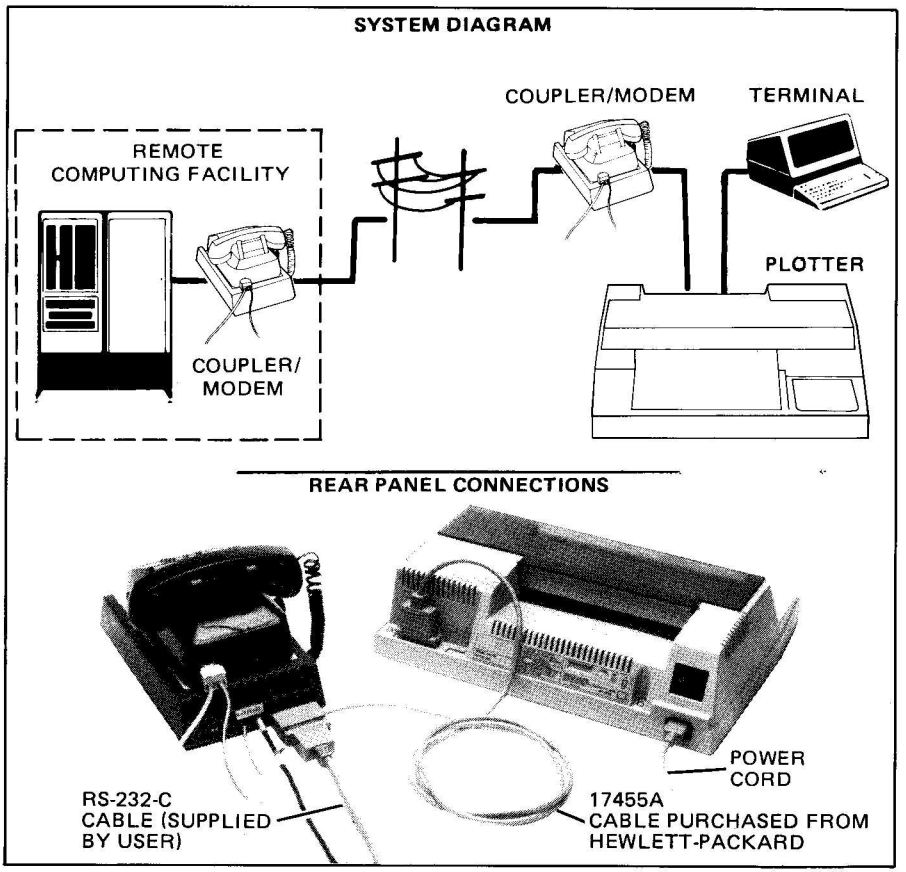
Operation with this type of installation is usually confined to the on-line, programmed-on state. The rear panel switch labeled **Y/D** should be set to **D** (direct). When the switch is set to **D**, whenever power is being applied to the plotter, it is in the on-line, programmed-on state. In this state, the plotter reacts to all device control and HP-GL instructions except the plotter off instruction. It is not possible to programmatically turn the plotter off. Only when the switch is set to **Y** may the plotter be

placed in the on-line, programmed-off mode. That operating state is described under operation with a terminal.

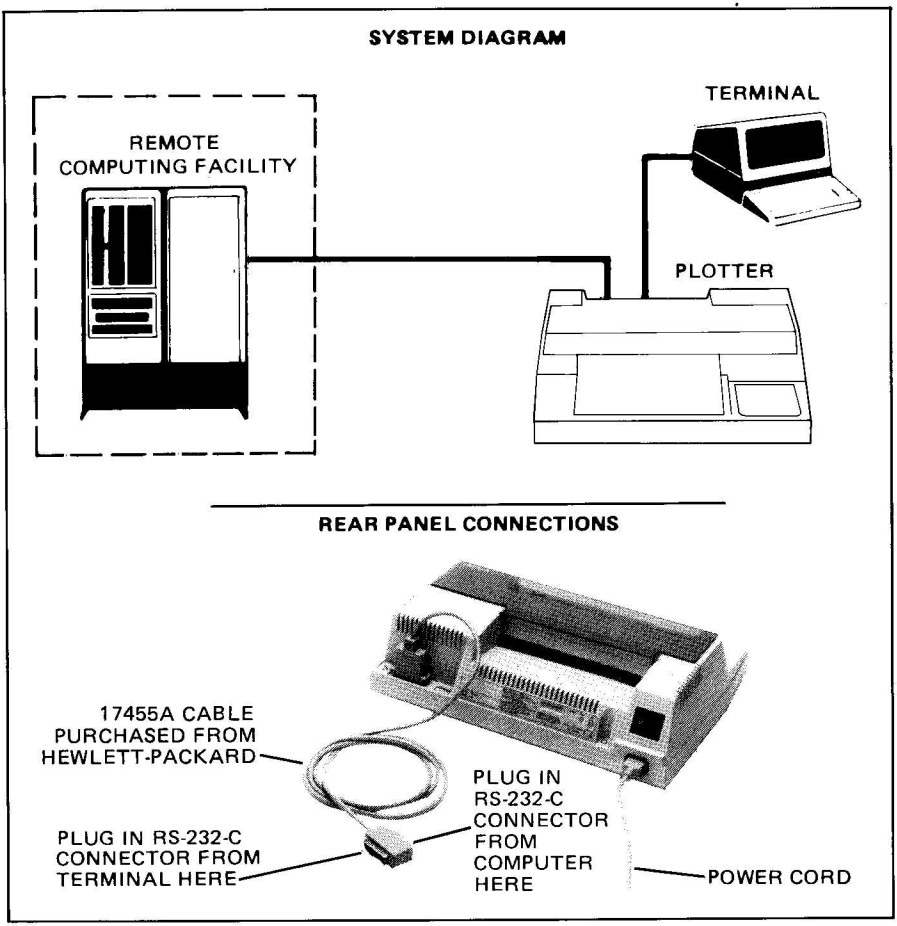
Using a Plotter in an Environment with a Terminal

Installation

In the second type of system, the plotter is connected in series between the computer and the terminal. The plotter's **LINE** switch must be **ON** in order to have any communication between the terminal and the computer. There may be a direct wire between the computer and the plotter or the plotter may be connected to a modem and communication may take place over telephone lines. This setup, with the plotter between the computer and the terminal, is sometimes referred to as eavesdrop environment. A special Y-cable (Part No. 17455A), which joins the lines from the computer and terminal into the plotter's one connector, must be used in this environment. Diagrams of the two systems, with and without a modem, follow, along with pictures of the rear-panel connections for both kinds of systems.



Plotter Interconnection with a Terminal and Remote Facility Using Modems



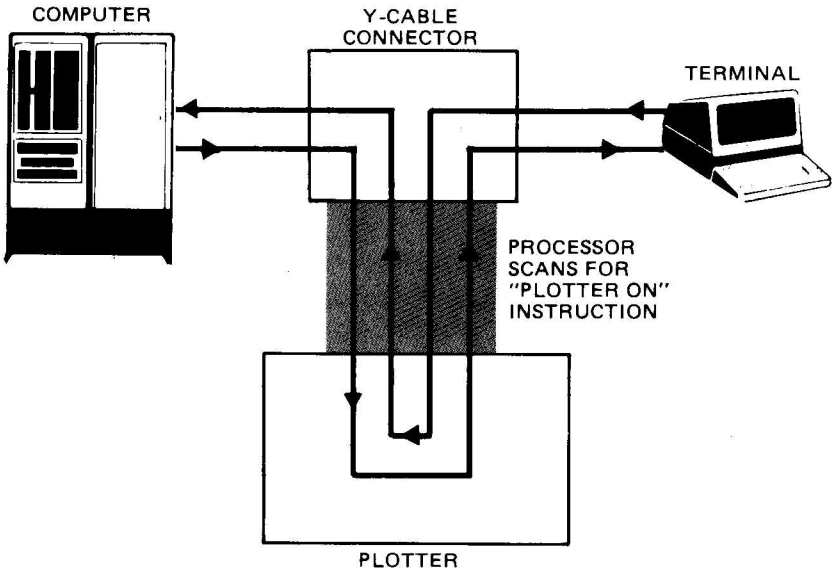
Plotter Interconnection with a Terminal and Remote Facility
Using RS-232-C/CCITT V.24 Cabling

Operation

While operating in this environment, the plotter may be in one of three states: on-line, programmed-off; on-line, programmed-on; or monitor mode.

On-Line, Programmed-Off State

The plotter can only be in this state if the **Y/D** switch on the rear panel is set to **Y** (used with Y-cable). When this switch is set to **Y**, the plotter is placed in the on-line, programmed-off state by either turning the plotter's **LINE** switch to **ON** or by receipt of a plotter off instruction from the computer or of a terminal-generated Break signal while the plotter is in the on-line, programmed-on state. In the on-line, programmed-off state, the plotter's processor passes data between the computer and the terminal as shown in the following diagram. The plotter will respond only to a plotter on instruction from the host computer.



Plotter in On-Line, Programmed-Off State

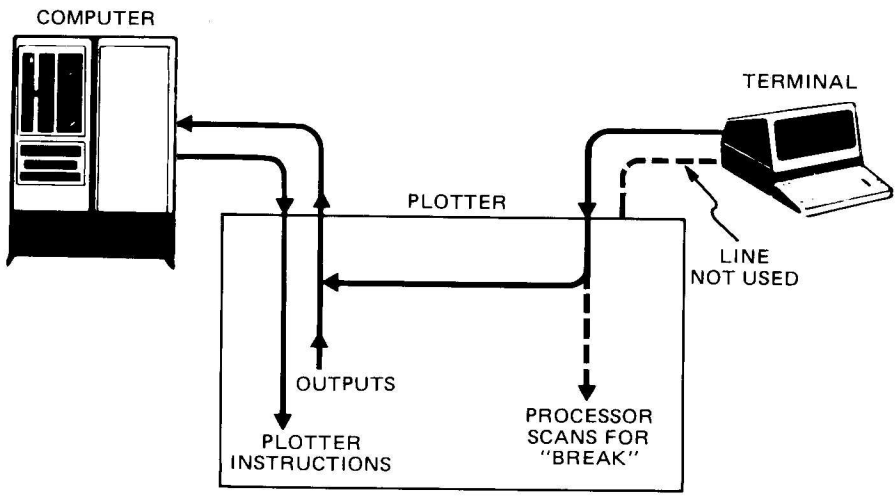
On-Line, Programmed-On State

When the rear-panel switch labeled **Y/D** is set to **D**, the plotter is placed in the on-line, programmed-on state by turning on the plotter. When the **Y/D** switch is set to **Y**, the plotter is switched from the on-line, programmed-off state to the on-line, programmed-on state when a plotter on instruction, ESC . (or ESC . Y, is received from the computer.

When in this state, the plotter operates in response to instructions received from the computer as shown in the following figure. 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 entry into 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 anything greater than a 130-millisecond space as a Break. This Break signal is retransmitted to the computer and in-process plotter outputs are aborted, but plotting continues until stored buffer data is completed. A new plotter on instruction from the computer is required to resume plotting operations. The plotter will ignore a Break signal if the **Y/D** switch is set to **D**.

It should be noted that in the on-line, programmed-on state (but not in monitor mode which is described in the next paragraph) all data generated by the terminal are routed through to the computer on a noninterference basis when the plotter is not doing outputs. Data generated by the terminal are ignored while output is occurring. However, all data generated by the computer are intercepted by the plotter and not passed to the terminal.

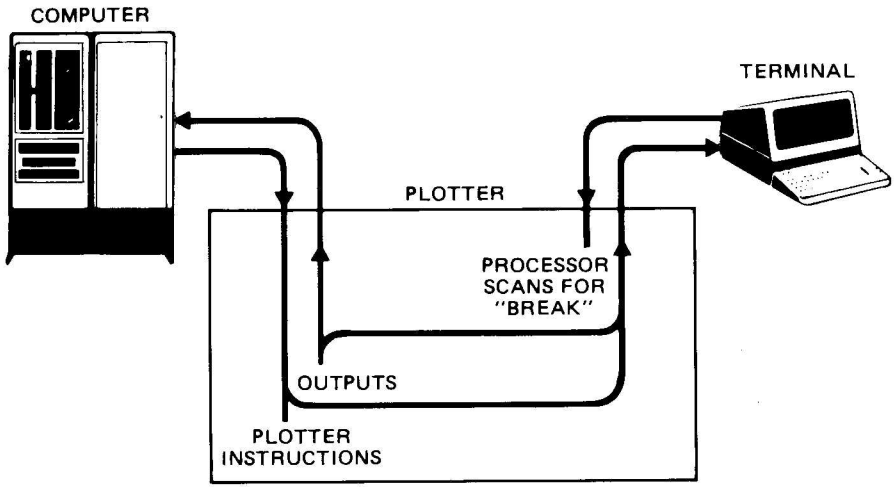


Plotter in On-Line, Programmed-On State

Monitor Mode

After the plotter is in the on-line, programmed-on state, two mutually exclusive monitor modes may be enabled using the set plotter configuration instruction, ESC . @. Depending upon which monitor mode is enabled, either all data (including device control instructions) are retransmitted to the terminal CRT or only HP-GL data are retransmitted as they are parsed from the plotter's buffer. All plotter output responses are sent to both the computer and terminal. Refer to The Set Plotter Configuration Instruction, ESC . @, for complete information.

The plotter monitors for a terminal-generated Break signal. Receipt of a Break signal will cause the same results as described under the on-line, programmed-on state. Then, new plotter on and set plotter configuration instructions from the computer are required to resume plotting operations with monitor mode active. The following diagram shows how the plotter processes data while in monitor mode.

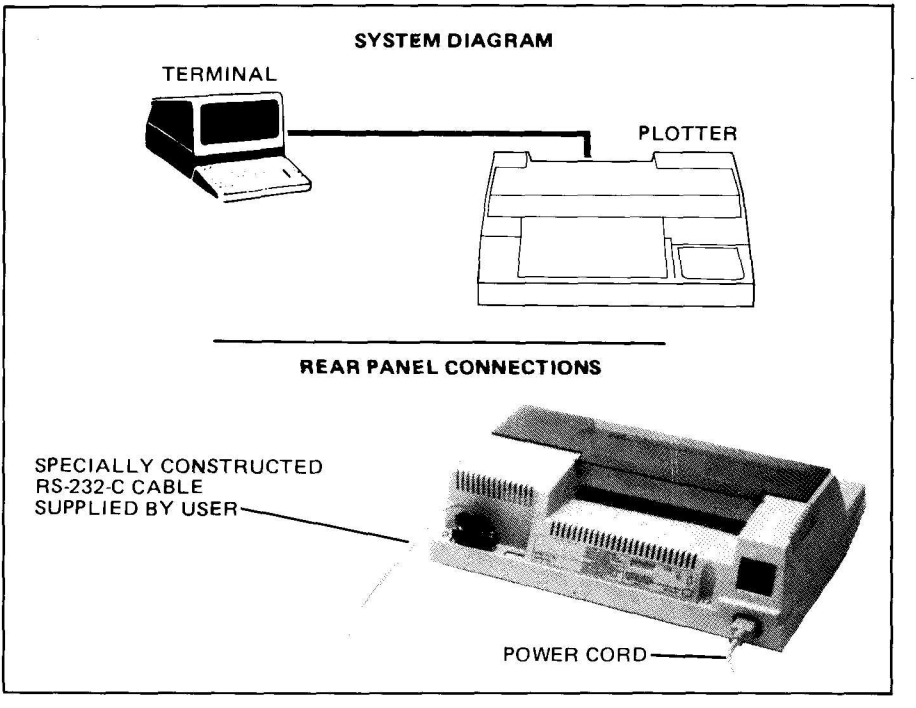


Monitor Mode

Using the Plotter in a Terminal-only Environment

Installation

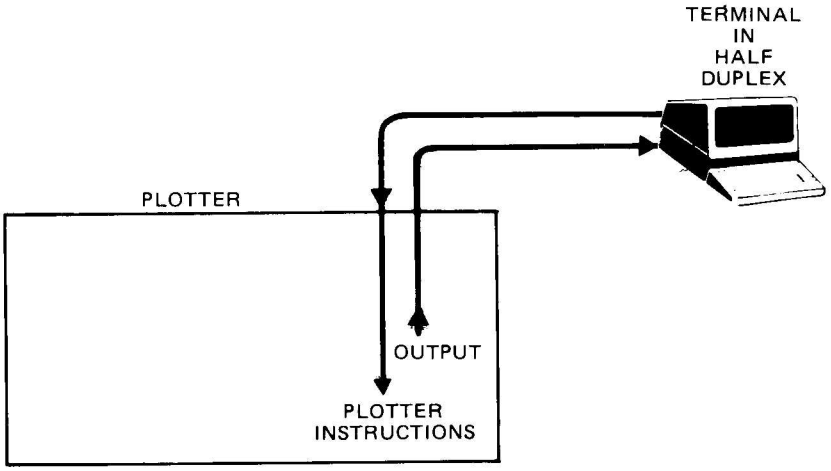
The 7470 plotter can be directly connected to a terminal if a specially constructed, user-supplied cable that swaps lines 2 and 3 is used. While there is no computer in this configuration, the terminal usually has some "intelligence." When the terminal and plotter are connected using this special cable, the terminal may be used to send instructions to the plotter. A diagram of the terminal-only environment and a picture showing the rear-panel connection follow.



Plotter Interconnection with Only a Terminal

Operation

The rear-panel switch labeled **Y/D** should be set to **D**. If it is set to **Y**, the plotter must receive a plotter on instruction, ESC . (or ESC . Y, before it will respond to other commands from the terminal. The terminal should be set to half duplex in order to view the characters being sent to the plotter. Plotter output will be displayed on the terminal. The following diagram shows plotter operation when in the programmed-on state in a terminal-only environment.



Terminal-only Environment, Programmed On

Connecting the RS-232-C Interface

The 7470 plotter interfaces to the RS-232-C communications lines through a standard 25-pin female connector mounted on the back of the plotter. The 7470 is capable of operating in a three-wire (transmit, receive, ground) configuration.

In hardwired handshake operation, the Data Terminal Ready line (pin 20 of the connector on the plotter) is used to monitor the space in the buffer available for input. The plotter outputs data when requested (refer to Hardwire Handshake in this chapter).

If you are fabricating the cable assembly, the connector should be a 25-pin type "D" subminiature CINCH DBC-25P plug or equivalent.

Connector pin allocations for the three-wire configuration are identified and described in the following table.

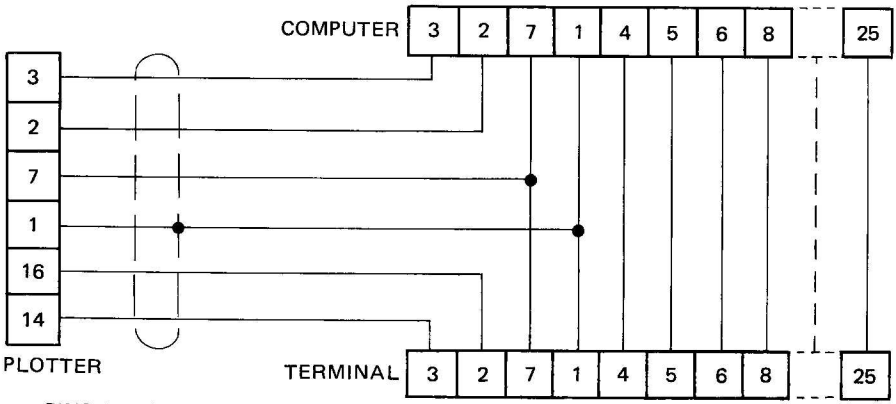
Minimum Interface Connector Pin Allocations

Pin No.	RS-232-C	CCITT V.24	Function/Signal Level
2	BA (TDATA)	103	Data line from plotter High = ON = "0" = +12 V = SPACE Low = OFF = "1" = −12 V = MARK
3	BB (RDATA)	104	Data line to plotter High = ON = "0" = +3 V to +25 V Low = OFF = "1" = −3 V to −25 V
7	AB (SGND)	102	Signal ground (Return line)

In addition to the minimum requirements for communication, six more lines are connected as shown in the following table. These lines are required to implement full duplex communication, intermediate baud rate, hardwired handshake mode, and monitor mode. All remaining pins make no internal connection.

Pins 14 and 16 are wired in the special Y-cable, available as Option 16, to implement monitor mode. The Y-cable schematic is shown below.

NOTE: Hardwire handshake **cannot** be used to prevent buffer overflow when the Y-cable is connected. This is because pin 20 is connected between the **COMPUTER** and **TERMINAL** connectors, but not to the **PLOTTER** connector. ■



Y-cable Schematic

Additional Connector Pin Allocations

Pin No.	RS-232-C	CCITT V.24	Function/Signal Level
1	AA	101	Protective ground
4	CA	105	Request To Send from the plotter Always High = ON = "0" ≅ +12 V
17	DD	115	External Clock Input High = ON = +2.4 V to +5 V Low = OFF = 0.0 V to +0.4 V
20	CD	108.2	Data Terminal Ready to modem High = ON = "0" ≅ +12 V Low = OFF = "1" ≅ −12 V
14*	SBA	118	Secondary Transmit Data Data line from plotter to terminal
16*	SBB	119	Secondary Received Data Data line to plotter from terminal

*Used to establish monitor mode with special Y-cable (Part No. 17455A).

Output Baud Rate

The plotter is designed to operate in an asynchronous mode with switch-selectable baud rates of 75, 110, 150, 200, 300, 600, 1200, 2400, 4800, and 9600. See the 7470A Operator's Manual for instructions on setting the baud rate. However, setting all **BAUD** switches to zero and connecting 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.

Stop Bits

The plotter is configured to automatically verify or generate one or two stop bits, depending on the setting of the plotter's baud rate switches. Refer to the 7470A Operator's Manual for more information.

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 party.

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).
- Overrun error — a plotter instruction writes over another instruction.
- Buffer overflow error — the plotter receives more bytes of data than it has space for in the buffer.

When the plotter detects a framing, parity, or overrun error, it turns on the front panel **ERROR** light and sets error code 15. This error code generally indicates that the communication incompatibility is hardware related (incorrect stop bit jumper installation, wrong parity selection, incompatible or incorrectly set baud rates, etc.).

When the plotter detects a buffer overflow, it turns on the front panel **ERROR** light and sets error code 16. The last HP-GL data that caused the overflow will be lost. Error code 16 generally indicates an improperly established handshake protocol.

The **ERROR** light remains on until either the user interrogates the plotter via an output extended error command, ESC . E, and the plotter responds with the appropriate error code, or the user turns the plotter off, or an HP-GL initialization instruction, IN, is processed, or a front-panel reset occurs.

A complete list of error codes is included with the discussion of the ESC . E instruction.

NOTE: A buffer overflow condition may also cause an HP-GL error to occur. In this case, an HP-GL IN or OE command or a front-panel reset must be executed in order to clear the **ERROR** light. See Chapter 7 for an explanation of the output error instruction, OE. ■

Handshaking

The 7470 uses a 255-byte input buffer to synchronize the processing of data with the rate at which it is received. The presence of an input buffer requires that the computer and the plotter transfer information to one another in such a way that data will not be lost or misinterpreted. This is the purpose of handshaking.

The 7470 is capable of using any one of four handshaking methods to prevent buffer overflow and the resulting loss of data. The computer system's capabilities and requirements dictate which handshake method is appropriate.

- Hardwire Handshake — uses a physical wire, pin 20 of the RS-232-C cable, to control handshaking. It can be used if the computer system can or does monitor pin 20 (DTR).
- Xon-Xoff Handshake — is managed by the peripheral device. It can be used if the computer system follows an Xon-Xoff protocol (control characters are transmitted from the peripheral to the computer).
- Enquire/Acknowledge Handshake — is managed by the computer system and interface. This handshake is often used in Hewlett-Packard systems and is so named because the ASCII characters ENQ and ACK may be used to control the handshake.
- Software Checking Handshake — is managed by the applications programmer. It can be used on almost any computer system, but it must be used if the system cannot implement any of the other three handshaking methods.

Once the handshake method is selected, the 7470 can be programmatically instructed to match the computer system requirements, implement the chosen handshake method, and function properly within the system-dependent communication environment. This is done by specifying certain variables in device control commands which are issued to the 7470 at the beginning of each computer session or graphics program. The variables, which may be specified by using the decimal value of the character desired to establish one of the four handshake methods available to the 7470, are:

- Output Trigger Character — The output trigger character, when used, is the last character output by the computer when making a request of a graphics peripheral. Defining this character in a command tells the plotter, "Don't respond to my request until you receive

this trigger character." This character is often a DC1 (decimal equivalent 17) or some other nonprinting ASCII character such as LF or CR or, when using some implementations of BASIC, the ? (decimal equivalent 63), which does print.

- Turnaround Delay — The turnaround delay is the length of time the plotter will wait after receiving a computer request and the trigger character, if any, 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 initator expect the start-of-text character, STX (decimal equivalent 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 equivalent 13), as the plotter's output terminator.

- Echo Terminate Character — Echoing is commonly found in full duplex systems. Use of the echo terminate parameter in a device control command tells the plotter that the computer will echo all responses and that this echoed data should be ignored (the plotter's data buffer should be closed) until an echo terminate character is received. When the plotter receives the echo terminate character, it reopens the data buffer to receive graphics data from the computer. Computers often use the line feed character, LF (decimal equivalent 10), as the echo terminator. If the computer does not echo the peripheral's response, this variable must be zero (equivalent to null) or must be omitted.

- Intercharacter Delay — Some computers cannot process data as fast as the plotter can transmit it due to limited buffering in the 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 Xon-Xoff handshake mode is to be established, a NULL character (decimal equivalent 0) must be specified as the enquiry character. If enquire/acknowledge is to be established, an ENQ character (decimal equivalent 5) or any other ASCII character besides the 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 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." Computers frequently require a DC3 character (decimal equivalent 19) for the immediate response.

- Acknowledgment String — The acknowledgment string specifies the character or characters that the plotter will send to the computer when the plotter's input buffer has room for another block of data. Computers frequently require that the ACK character (decimal equivalent 6) be used for the acknowledgment string.

- Data Block Size — This is the maximum size of each data block the computer will transmit to the plotter.

- Data Terminal Ready (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 mode. Pin 20 is held on (+12 V) if hardwire handshake is disabled.

- Xoff Threshold Level — In the Xon-Xoff handshake mode this defines how many empty bytes remain in the buffer when the plotter sends the Xoff trigger character to the computer, telling it to stop sending data.

- 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 equivalent 19) is generally used for the Xoff trigger.

- Xon Trigger Character — This specifies the character string the plotter will use to signal the computer that there is sufficient space in the buffer to resume sending data. The DC1 character (decimal equivalent 17) is generally used for the Xon trigger.

The following discussion of the four handshake methods includes the pertinent variables and identifies the commands which will establish their values.

Software Checking

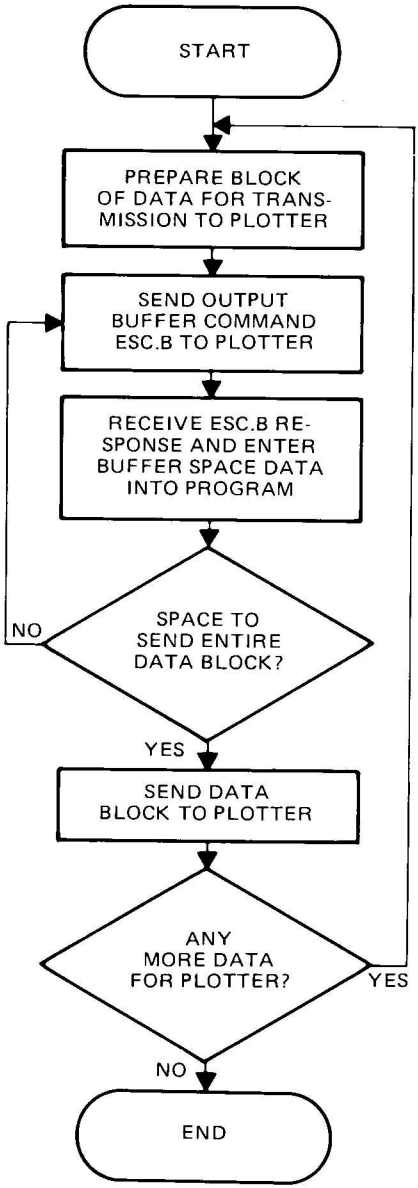
Software checking is a nonautomatic handshake method in which the user's program repeatedly asks the plotter how many characters of empty space remain in the buffer. When the plotter response is bigger than the next block of data, the program will transmit the data block to the plotter. This method is inefficient in time-share environments.

The advantage of software checking is that it is independent of hardware and operating system abilities required to implement other handshake modes; therefore, it usually makes software transportable between computer systems. The limitation of this method of handshaking is that it uses up computer time.

To match the requirements of the computer system, these variables may be specified for the software checking handshake mode by using the appropriate command:

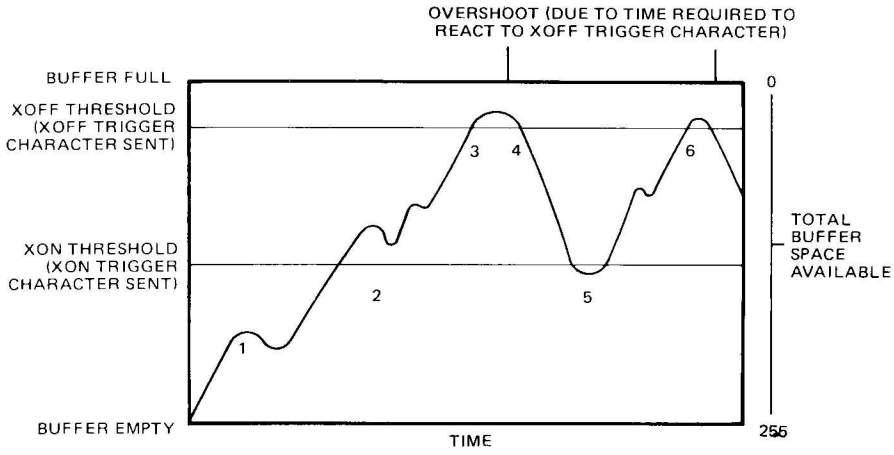
- Turnaround delay (ESC . M command)
- Output trigger character (ESC . M command)
- Echo terminate character (ESC . M command)
- Output initiator character (ESC . M command)
- Output terminator (ESC . M command)
- Intercharacter delay (ESC . N command)

The following flow diagram illustrates the functional elements of a typical software checking handshake within a user's program.



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 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 data is sent to the plotter by the computer, it is stored in the buffer and simultaneously acted on 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 overshoot may occur. For this reason, the Xoff threshold level should always be specified at least as large as the data block size or the

maximum number of bytes sent by an output statement to allow room for the overshoot.

5. Once the Xoff trigger character has been sent, 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 at 128 bytes. If the Xoff threshold level is greater than 128, the Xon threshold is reset to send the Xon character when one more byte than required by the Xoff threshold is available in the plotter's buffer.
6. Data is again stored in the buffer until all the data are transferred or until the Xoff threshold level is exceeded again.

The following conditions can be specified for the Xon-Xoff handshake mode to match the requirements of the computer system, by using the appropriate command:

- Xoff threshold level (ESC . I command)
- Xon trigger character (ESC . I command)
- Xoff trigger character (ESC . N command)
- Intercharacter delay (ESC . N command)

The enquiry character (ESC . I command) must either be defaulted or specified as zero.

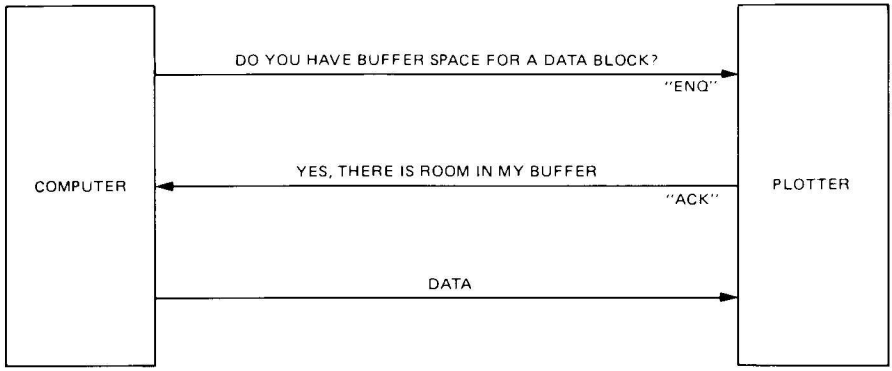
Enquire/Acknowledge Handshake

With the enquire/acknowledge handshake, the computer's operating system or application program initiates the data exchange process by querying the plotter about the availability of buffer space. The format of the exchange is dependent upon the requirements of the computer. The following conditions can be specified for the enquire/acknowledge handshake mode by using the appropriate command:

- Turnaround delay (ESC . M command)
- Output trigger character (ESC . M command)
- Echo terminate character (ESC . M command)
- Output initiator character (ESC . M command)
- Output terminator (ESC . M command)
- Intercharacter delay (ESC . N command)
- Immediate response string (ESC . N command)
- Data block size (ESC . I or ESC . H command)

- Enquiry character (ESC . I or ESC . H command)
- Acknowledgment string (ESC . I or ESC . H command)

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 commands 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 = ASCII character DC1 (decimal equivalent 17)

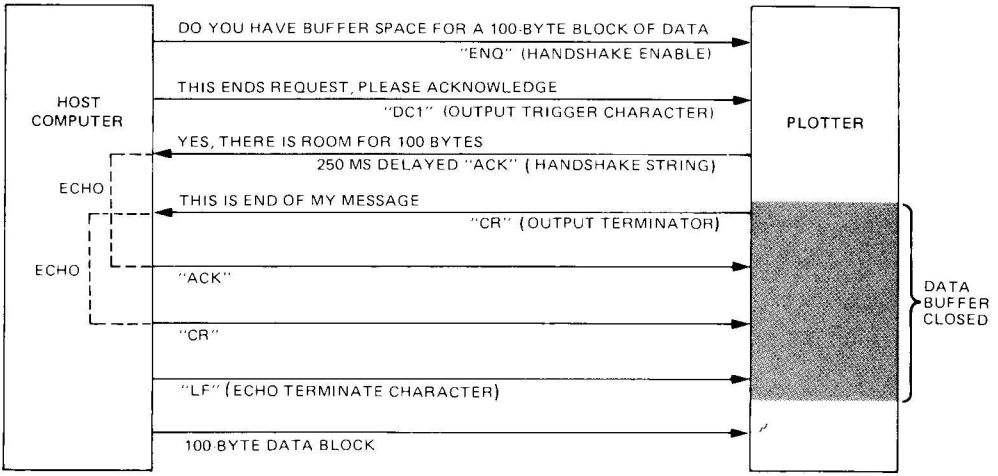
echo terminate character = ASCII character LF (decimal equivalent 10)

output terminator = ASCII character CR (decimal equivalent 13)

data block size = 100 bytes

enquiry character = ASCII character ENQ (decimal equivalent 5)

acknowledgment string = ASCII character ACK (decimal equivalent 6)



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 sequence 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. If there is enough room in the plotter's buffer to accept and store another block of data, the plotter sets the Data Terminal Ready, CD, line to a high state. If there is insufficient space, it sets the line low. By monitoring this line, the computer knows when it can or cannot safely transmit another block of data.

The hardwire handshake mode is enabled at power on or by setting the Data Terminal Ready, CD, line control using the ESC . @ command.

RS-232-C Device Control Instructions

Device control instructions establish the handshake protocol to be used by the 7470 plotter. All communications conform to the protocol established by these instructions. The instructions serve two purposes: to control the format by which data is transferred between the computer and the plotter (input/output operations), and to give the computer the ability to query and to receive information from the plotter.

Each instruction's name gives an immediate clue to its purpose: if "output" is the first word in the name of the instruction, the computer wants a response from the plotter. Otherwise, the instruction concerns the I/O functions. The word "set" in the title indicates the command establishes conditions under which subsequent I/O is to occur.

The plotter acts on device control instructions immediately upon receipt. It does not store them in the data buffer.

Command Syntax for Device Control Instructions

Device control instructions are three-character escape code sequences comprised of "ESC" and "." followed by one of the characters @, B, E, H, I, J, K, L, M, N, or O, R, (,), Y, or Z.

When an instruction is put together with its required parameters, delimiters, and/or terminators, it becomes a "command." These syntax conventions are used with the commands discussed in this chapter:

[ ]	Brackets indicate that all parameters enclosed are optional.
( )	Parentheses indicate that each individual parameter is optional.
;	The semicolon follows and delimits parameters. If a semicolon appears without a parameter, the parameter is defaulted.
:	The colon terminates any command which may have parameters and can occur after any valid number of parameter entries. Any parameter that is not specified is defaulted.
<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 equivalent for an ASCII character (see the ASCII Character Equivalents table in Appendix C). In this case, the characters 10 would represent the ASCII line feed character, LF, and 13 would represent the ASCII carriage return character, CR.
. . .	Specifies a number of optional parameters. Each parameter must be followed by a delimiter (;) or the terminator (:).
[TERM]	Unless changed by an ESC . M command, all RS-232-C output responses include a CR as a terminator.

Default Values; Omitting Parameters — Any parameter may be omitted or, if the parameter is required, it can be set to its default value by omitting the parameter and entering only the semicolon as a delimiter. All parameters may be omitted and therefore set to default values by entering only the colon terminator after the instruction.

ESC — Denotes the single ASCII character, Escape, which in most computers is accessed by striking a single key on the keyboard.

NOTE: There is no delimiter (semicolon) between the three-character command sequence, e.g., ESC . O, and the first parameter. ■

The Plotter On Instruction, ESC . (or ESC . Y

DESCRIPTION The plotter on instruction, ESC . (or ESC . Y, places a plotter which is powered on into the on-line, programmed-on mode so that it will accept incoming data and interpret it as plotter instructions.

USES This instruction is used when the rear-panel switch labeled Y/D is set to Y to ready the plotter to accept other instructions. It is sent at the beginning of any plotting program or when the user wishes to resume plotting after the plotter has been turned off by an ESC .) or ESC . Z command or a Break.

SYNTAX ESC . (
or
ESC . Y

EXPLANATION This instruction is ignored when the rear-panel switch labeled Y/D is set to D since, in that case, turning on the power places the plotter in the programmed-on state.

Beginning with the next character, the plotter will accept incoming data and interpret it as plotter instructions. If the plotter is already in the programmed-on state, it will ignore this instruction.

The Plotter Off Instruction, ESC .) or ESC . Z

DESCRIPTION The plotter off instruction, ESC .) or ESC . Z, takes the plotter out of on-line, programmed-on state so that it neither accepts nor interprets incoming data until another plotter on instruction is received.

USES The instruction is used to deactivate the plotter. It is used at the end of a graphics program or in some environments to allow data to be passed through the plotter to the terminal.

SYNTAX ESC .)
or
ESC . Z

EXPLANATION This instruction is ignored when the rear-panel switch labeled Y/D is set to D. When that switch is set to D, it is not possible to turn the plotter off programmatically.

Beginning with the next character, the plotter will assume a passive state and remain in that state until a plotter on instruction is received.

Any HP-GL instructions remaining in the buffer at the time that a plotter off instruction is received are executed. However, no additional HP-GL instructions will be accepted by the plotter.

NOTE: A Break signal from the terminal will have the same effect as a plotter off instruction. ■

The Set Plotter Configuration Instruction, ESC . @

DESCRIPTION The set plotter configuration instruction, ESC . @, sets parameters necessary for hardwire handshake mode and monitor mode.

USES The instruction is used to enable or disable hardwire handshake or monitor mode.

SYNTAX ESC . @ [(<DEC>) ; (<ASC>)] :

DEFAULT ESC . @ : Enables hardwire handshake and disables monitor mode.

EXPLANATION Use of the instruction without parameters enables hardwire handshake and disables monitor mode.

A description of the instruction's parameters follows:

<DEC> The first parameter is not required; if a parameter is included it is ignored. The semicolon must precede any second parameter.

<ASC> The second parameter establishes Data Terminal Ready, CD, line control. Only bits 0, 2, and 3 of the parameter are used, as shown in the following table.

Bit No.	Logic State	Description
0	0	Set and hold line high (disable hardwire handshake).
	1	Enable hardwire handshake mode.*
1	X	Ignored.
2	0	Establish monitor mode 0 (all bytes displayed on terminal as they are parsed from the buffer).
	1	Establish monitor mode 1 (all bytes displayed as they are received).
3	0	Disable monitor mode.
	1	Enable the monitor mode established by bit 2.

*When hardwire handshake is enabled, the DTR line becomes a "buffer space available" flag. The line is high when available buffer space is greater than or equal to the current block size, and is held low when available buffer space is less than the current block size. This size defaults to 80 bytes unless a different value is specified by the ESC . H or ESC . I command.

EXAMPLE ESC . @;13: will establish monitor mode 1 where all bytes are displayed on the terminal as they are received by the plotter.

The Output Buffer Space Instruction, ESC . B

DESCRIPTION The output buffer space instruction, ESC . B, outputs the plotter's available buffer space.

USES This command is used in a software checking handshake to interrogate the plotter regarding available buffer space.

SYNTAX ESC . B

EXPLANATION No parameters are used.

RESPONSE

<DEC> The plotter's response is a decimal number in the range 0 to 255, and represents the number of bytes of buffer space currently available for storing graphic instructions sent from the computer.

[TERM] This decimal number is followed by the output terminator which defaults to carriage return, CR, or is as set by ESC . M.

The Output Extended Error Instruction, ESC . E

DESCRIPTION The output extended error instruction, ESC . E, outputs a number which defines any RS-232-C related I/O error and turns off the front-panel **ERROR** light.

USES The instruction is used to define what type of RS-232-C related I/O error has occurred, if any.

SYNTAX **ESC** . E

EXPLANATION No parameters are used.

RESPONSE

<DEC> The plotter's response is a decimal number, either 0 or in the range 10-16, followed by the output terminator. The meaning of the response is as defined in the following table.

Error No.	Meaning
0	No I/O error has occurred
10	Output instruction received while another output instruction is executing. The original instruction will continue normally; the one in error will be ignored.
11	Invalid byte received after first two characters, **ESC** ., in a device control instruction.
12	Invalid byte received while parsing a device control instruction. The parameter containing the invalid byte 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 exit) or the first byte of another instruction is received (abnormal exit).
15	A framing error, parity error, or overrun error has been detected.
16	The input buffer has overflowed. As a result, one or more bytes of data have been lost, and therefore an HP-GL error will probably occur.

NOTE: The receipt of something other than another parameter, a semicolon, or a colon will result in error 12 overwriting error 14. ■

[TERM] The terminator defaults to carriage return, CR, unless it is set by an ESC . M.

The Set Handshake Mode 1 Instruction, ESC . H

DESCRIPTION The set handshake mode 1 instruction, ESC . H, may be used with the enquire/acknowledge or Xon-Xoff handshake to establish parameters for the plotter's communication format.

USES It establishes the data block size, the enquiry character, and the acknowledgment string when the computer requires that the parameters set in the ESC . M instruction be used in response to the enquiry character or Xon character.

SYNTAX ESC . H [(<DEC>) ; (<ASC>) ; (<ASC>(; . . . <ASC>))] :

DEFAULT ESC . H : See ESC . I default.

EXPLANATION The two instructions, ESC . H and ESC . I, are mutually exclusive. The parameter descriptions are the same for both instructions and are given under the ESC . I instruction.

Handshake mode 1, established by this command, uses defaulted or specified parameters of the ESC . M and ESC . N commands when responding to the handshake enable or Xon trigger character.

The parameters used with handshake mode 1, handshake mode 2, and output responses are shown in the following table. Choose the mode and use handshake mode 1 (ESC . H) or handshake mode 2 (ESC . I) depending on the requirements of your system.

Parameter Usage in Plotter/Computer Communication

Parameter	With Handshake Characters		With Plotter Output Commands
	In Mode 1	In Mode 2	
turnaround delay	yes	yes	yes
output trigger character	yes	no	yes
echo terminator	yes	no	yes
output terminator	yes	no	yes
output initiator*	no	no	yes
intercharacter delay	yes	yes	yes

*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.

EXAMPLES See ESC . I and ESC . N.

The Set Handshake Mode 2 Instruction, ESC . I

DESCRIPTION The set handshake mode 2 instruction, ESC . I, may be used with the enquire/acknowledge or Xon-Xoff handshake to establish parameters for the plotter's communication format.

USES It establishes the data block size, the enquiry character, and the acknowledgment string for the enquire/acknowledge handshake when the computer expects only the turnaround delay, and not the other parameters set by ESC . M, to be included in the response to the enquiry character. It sets the Xoff threshold level and the Xon trigger character for Xon-Xoff handshake.

SYNTAX ESC . I [(<DEC>) ; (<ASC>) ; (<ASC>(; . . . <ASC>))] :

DEFAULT ESC . I : (or ESC . H :) Neither Xon-Xoff nor enquire/acknowledge handshake is enabled. 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 buffer overflow.

EXPLANATION The two instructions, ESC . I and ESC . H, are mutually exclusive. With handshake mode 2, the only parameter of the ESC . M command used when responding to the enquiry or Xon trigger character is the turnaround delay. Refer to the chart under the ESC . H instruction to see which parameters are used in various plotter output situations. Choose your mode using ESC . I or ESC . H, depending on the requirements of your system.

The parameters for both ESC . H and ESC . I are the same and are described below, first as interpreted for the enquire/acknowledge handshake and then as interpreted for the Xon-Xoff handshake.

For Enquire/Acknowledge Handshake

<DEC> This first parameter specifies the **block size;** it is evaluated modulo 256. 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 equivalent of any ASCII character in the range 0 to 127. 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 (enquire character, ENQ) is generally used.

<ASC>...<ASC> This is a list of 1 to 10 parameters, separated by semicolons, which specify the **acknowledgment string.** Decimal equivalents of ASCII characters 0 to 127 are valid. The value 0 is not transmitted and will terminate the string. The value 6 (acknowledge character, ACK) is generally used. If the parameter is omitted, it assumes its default value and no characters are sent.

For Xon-Xoff Handshake

<DEC> This first parameter sets the **Xoff threshold level** by specifying the number of empty bytes remaining in the buffer when the Xoff character is to be sent. The practical range is 10 to 254. If the Xoff parameter is specified to be greater than 128 (half the buffer size), the Xon threshold level will be reset (from its automatic setting of half the buffer size) so that the Xon character will be sent when one byte more than the Xoff level is available.

<ASC> This parameter should be omitted by entering only the 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 be included.

<ASC>...<ASC> This is a list of from 1 to 10 parameters, separated by semicolons, which specify the **Xon trigger character(s).** Decimal equivalents of ASCII characters 0 to 127 are valid. The value 0 is not transmitted and will terminate the string.

EXAMPLES See also the ESC . N instruction.

For 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 Bell, as the acknowledgment string. Since ESC . H sets handshake mode 1, the currently defined output initiator, output terminator, output trigger character, and echo terminator, as well as both turnaround delay and intercharacter delay, are used when the response string, DC4 Bell, is sent.

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 single ASCII character ACK as the acknowledgment string. Only the turnaround delay, intercharacter delay, and immediate response string, if any, are used when sending the response. No output initiator will precede it, even if one is defined, and no output terminator will follow it.

For Xon-Xoff Handshake

ESC . I81;;17: will set the Xoff threshold level to 81 (the Xoff character will be sent when 81 empty bytes remain in the plotter's buffer) and set the Xon trigger character to DC1. The second parameter is defaulted as required for this handshake. The Xoff trigger character must be set using the ESC . N command. Transmittal of the Xon and Xoff trigger characters is subject only to turnaround and intercharacter delays, if any are specified. No output initiator will precede them, even if one is defined, and no output terminator will follow them.

The Abort Device Control Instruction, ESC . J

DESCRIPTION The abort device control instruction, ESC . J, aborts any device control instruction that may be partially decoded or executed.

USES This instruction may be used in an initialization sequence when you first access the plotter.

SYNTAX ESC . J

EXPLANATION This instruction aborts any single device control instruction that may be partially decoded or executed. Unspecified parameters of aborted instructions are defaulted. All pending or partially transmitted output requests, from either HP-GL or device control instructions, are immediately terminated, including output responses and handshake parameters. Intermediate output operations such as turnaround delay and echo suppression are aborted, and the buffer input is enabled. The handshake and output mode parameters remain as specified.

The Abort Graphic Instruction, ESC . K

DESCRIPTION The abort graphic instruction, ESC . K, aborts any partially decoded HP-GL instruction and discards instructions in the buffer.

USES The instruction can be used as part of an initialization sequence when starting a new program or to terminate plotting of HP-GL instructions in the buffer.

SYNTAX ESC . K

EXPLANATION Any partially decoded HP-GL instruction is aborted and all instructions in the buffer are discarded. A partially executed instruction is allowed to finish.

The Output Buffer Size Instruction, ESC . L

DESCRIPTION The output buffer size instruction, ESC . L, outputs the size, in bytes, of the plotter's buffer.

USES The instruction is used to obtain information on the size of the plotter's buffer. This information might be used to determine parameters of commands which set up handshaking.

SYNTAX ESC . L

EXPLANATION No parameters are used. The instruction causes the 7470 to output, in ASCII, a decimal number equal to the number of bytes in the plotter's buffer.

RESPONSE

<DEC>	255
[TERM]	Defaults to carriage return, CR, or is as set by ESC . M.

The Set Output Mode Instruction, ESC . M

DESCRIPTION The set output mode instruction, ESC . M, establishes parameters for the plotter's communication format.

USES The instruction is used to establish a turnaround delay, an output trigger character, an echo terminate character, and an output initiator character. It is also used 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 equivalent 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 colon must be used following the last parameter (if any). Use of the instruction without parameters is equivalent to ESC . M: (see DEFAULT).

A description of the instruction's parameters follows.

<DEC> The first parameter is optional. If present, it is the **turnaround delay.** The delay implemented is ((parameter × 1.1875)mod 65 536)/1.2 milliseconds. The parameter range is 0 to 54 612. If parameters follow, the semicolon must be included even if this decimal parameter is omitted.

<ASC> The second parameter is also optional and, if omitted, assumes its default value of 0 (no trigger character). If included, it specifies a single character which becomes the **output trigger character.** The parameter may be the decimal equivalent of any ASCII character in the range 0 to 127. If parameters follow, the semicolon must always be included, even when this parameter is omitted.

<ASC> The third parameter is optional and, if omitted, assumes its default value 0 (no echo terminate character). If included, it specifies a single character which becomes the **echo terminate character.** The parameter may be the decimal equivalent of any ASCII character in the range 0 to 127. If parameters follow, the semicolon must always be included, even when this parameter is omitted.

<ASC> . . . <ASC> The fourth parameter is optional and defaults to 13, the decimal equivalent of the single ASCII character, carriage return.

If included, the parameter may be the decimal equivalent(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 a parameter follows, the semicolon must always be included, even when this parameter is omitted. If the fifth parameter is specified, this fourth parameter must consist of two characters, or the second character must be specified as null using the semicolon.

<ASC> The fifth parameter is optional and, if omitted, assumes its default value 0 (no output initiator character). If included, it is the decimal equivalent of a single character which becomes the **output initiator** character. The parameter may be the decimal equivalent of any ASCII character in the range 0 to 127. The parameter is followed by a colon.

EXAMPLES See the ESC . N instruction.

The flowchart on the next page depicts plotter output.

The Set Extended Output and Handshake Mode Instruction, ESC . N

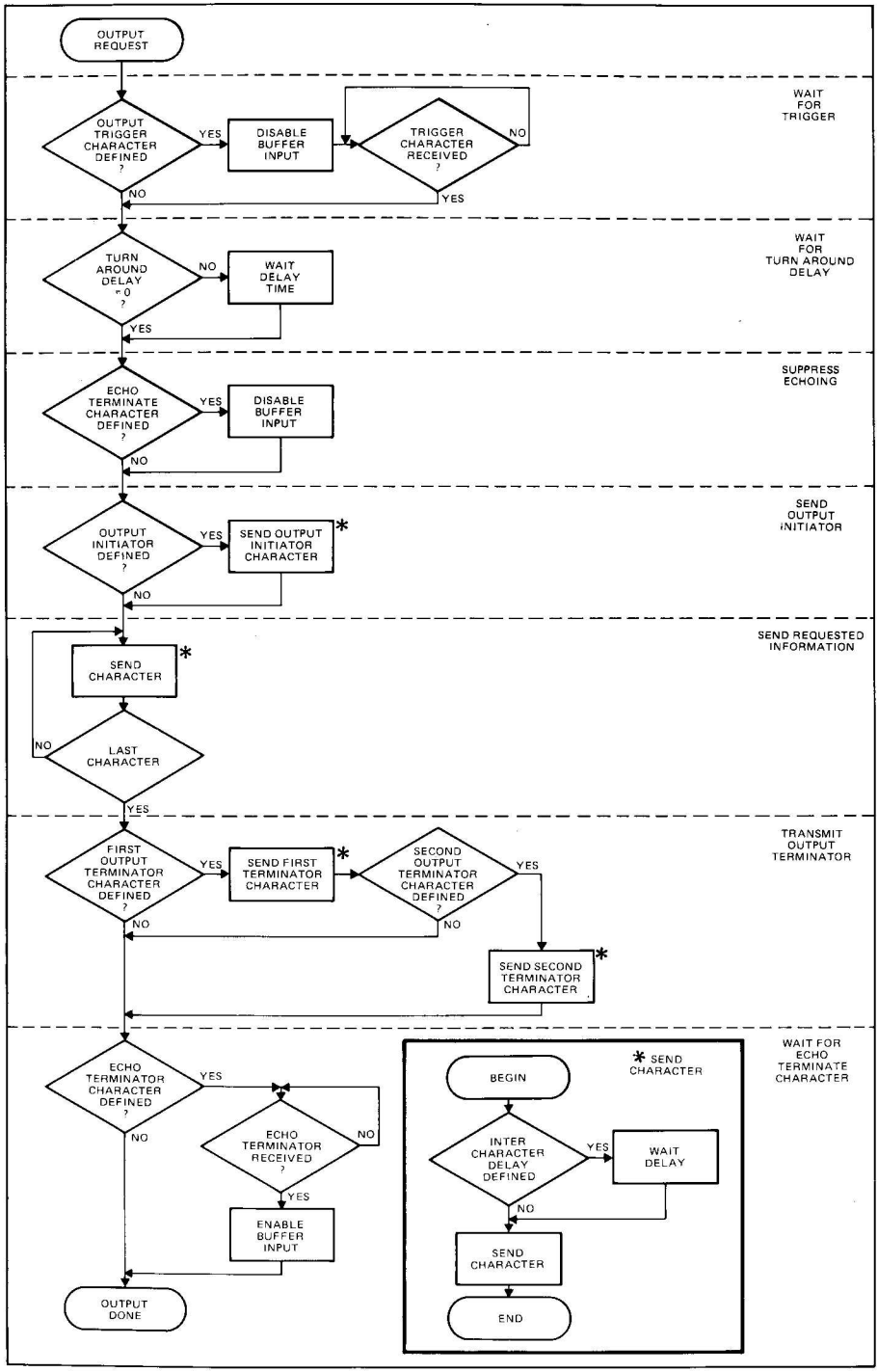
DESCRIPTION The set extended output and handshake mode instruction, ESC . N, establishes parameters for the plotter's communication format.

USES The instruction is used to specify an intercharacter delay in all handshake modes, the immediate response string for 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 colon must be used following the last parameter. Use of the instruction without parameters is equivalent to ESC . N : (see DEFAULT).



Output Request Flow Chart

A description of the instruction's parameters follows:

<DEC> The first parameter is optional. If present, it is the **intercharacter delay.** The delay implemented is ((parameter × 1.1875)mod 65 536)/1.2 milliseconds. The parameter range is 0 to 54 612. If parameters follow, the semicolon must be included, even if this decimal parameter is omitted.

<ASC>...<ASC> This parameter is optional. If present, it is a list of the decimal equivalents of 1 to 10 ASCII characters in the range 0 to 127. For **Xon-Xoff handshake mode,** it specifies the **Xoff trigger character(s).** For **enquire/acknowledge handshake mode,** it specifies the **immediate response string.** Semicolons must separate each parameter in the list.

EXAMPLES

For Xon-Xoff Handshake

ESC . N;19: Sets the Xoff trigger character to DC3. There will be no intercharacter delay, since the first parameter is defaulted to zero by the semicolon.

For Enquire/Acknowledge Handshake

The examples given here include all handshaking instructions. In addition to illustrating the use of intercharacter delays and immediate response strings set by ESC . N, they are designed to clarify the difference between handshake mode 1 and mode 2 and give some insight into why certain values are logical choices for some parameters. Note the CHR$ function is used to send the escape character.

```
10 DIM OUT$(80)
.
.
40 PRINT CHR$(27);".M0;63;0;13:";CHR$(27);".N5:"
50 PRINT CHR$(27);".H80;18;49:"
60 OUT$="IN;SP1;PA500,500;":GOSUB 100
.
.
.
100 PRINT CHR$(18): INPUT Z: PRINT OUT$: RETURN
```

The following parameters are set in lines 40 and 50:

turnaround delay = 0,

output trigger character = ? (decimal equivalent 63),

no echo terminate character,

output terminator = carriage return (decimal equivalent 13),

intercharacter delay = 5,

no immediate response string,

block size = 80,

enquiry character = DC2 (decimal equivalent 18), and

acknowledgment string = 1 (decimal equivalent 49).

The subroutine in line 100 contains the handshake. It causes the following chronological action. The enquiry character, DC2, is sent asking if the plotter has room for an 80-byte block. The plotter does not send an immediate response because that has been specified as null by its omission in the ESC . N command. The plotter holds its response until after it receives the output trigger character, ?. The question mark is sent by the computer when it interprets the BASIC statement INPUT to prompt for the input, Z. Z is the variable into which the acknowledgment string, 1, is read. If the acknowledgment string had been specified to contain nonnumeric characters, a string variable such as Z$ would have been used instead of Z.

The plotter waits approximately five milliseconds, the intercharacter delay, before sending the 1 and between the 1 and the output terminator, carriage return. Note the carriage return parameter could have been omitted, but carriage return still would have been sent as the output terminator because that is the default value for output terminator. If ESC . I had been used instead of ESC . H, the output terminator would not have been sent after the acknowledgment string (but it would follow responses to HP-GL output commands). The carriage return character is a logical choice, because it is expected by the computer to delineate the end of data read by the INPUT statement.

The computer is now free to send the string OUT$, which contains HP-GL commands, to the plotter. Note the enquiry character must be sent each time data is sent to the plotter.

Another handshake which would work using ESC . I is

```
40 PRINT CHR$(27);".I80;7;33;13:"
50 PRINT CHR$(27);".M500:";CHR$(27);".N5:"
.
.
.
100 PRINT CHR$(7):INPUT Z$: PRINT OUT$: RETURN
```

The following parameters are established:

turnaround delay = 500,

no output trigger character,

no echo terminate character,

output terminator = default value, carriage return,

intercharacter delay = 5,

no immediate response string,

block size = 80,

enquiry character = bell (decimal equivalent 7), and

acknowledgment string = ! carriage return (decimal equivalent 33, 13)

Now the computer sends the Bell character as the enquiry character. The plotter waits approximately 505 milliseconds, the total of the turnaround delay and the intercharacter delay, before sending its response. During that time, the computer will send the ? due to the INPUT statement, but the plotter ignores it. The plotter response to the enquiry character is now two characters, ! followed by a carriage return. The carriage return to terminate INPUT is now part of the acknowledgment string. No output terminator, now defaulted to carriage return, is sent because handshake mode 2 is set here by ESC . I. The output terminator, carriage return, will still follow all responses to HP-GL output commands.

The Output Extended Status Instruction, ESC . O

DESCRIPTION The output extended status instruction, ESC . O, outputs the plotter's extended status, giving information about the state of the buffer, pinch wheels, and **VIEW** button.

USES The instruction can be used to determine, from a remote location, if the plotter is ready to plot.

SYNTAX **ESC** . O

EXPLANATION No parameters are used. Unlike the HP-GL output status instruction, OS, the ESC . O instruction does not enter the buffer but is executed immediately, subject to any turnaround or intercharacter delays specified by ESC . M and ESC . N.

RESPONSE

<DEC> The response is the decimal equivalent of a 16-bit immediate status word, followed by the output terminator. The maximum value output is 40.

The extended status word bits are as defined in the following table.

Bit	State	Decimal Value	Meaning
0-2	0	0	Not used, always zeros. Reserved for plotters with paper advance.
3	0	0	Buffer is not empty.
	1	8	Buffer is empty and ready for data.
4, 5	00	0	Ready to process or processing HP-GL instructions.
	01	16	Paper loaded, **VIEW** button pressed so graphics suspended.
	10	32	Paper lever raised so graphics suspended.

Combinations of these bits allow five different responses to the ESC . O instruction.

Response	Meaning
0	Buffer is not empty and plotter is processing HP-GL instructions.
8	Buffer is empty and is ready to process or is processing HP-GL instructions.
16	Buffer is not empty and **VIEW** has been pressed.
24	Buffer is empty and **VIEW** has been pressed.
32	Buffer is not empty and paper lever and pinch wheels are raised.
40	Buffer is empty and paper lever and pinch wheels are raised.

[TERM] The output terminator defaults to carriage return unless it is set by ESC . M.

The Reset Handshake Instruction, ESC . R

DESCRIPTION The reset handshake instruction, ESC . R, resets all handshake parameters to their default values.

USES The instruction may be used to set the plotter's handshake responses to a known state with hardwire handshake enabled.

SYNTAX ESC . R

EXPLANATION Executing this command is the same as executing the following commands without parameters: ESC . @, ESC . H, ESC . I, ESC . M, and ESC . N.

The following table shows the default values of parameters used to establish handshakes.

Parameter	Value
block size	80
enquiry character	0 — no enquiry character
acknowledgment string	0 — no acknowledgment string
turnaround delay	0 — no delay
output trigger character	0 — no trigger character
echo terminate character	0 — no echo terminate character
output terminator	13;0; — carriage return
output initiator	0 — no output initiator
intercharacter delay	0 — no delay
immediate response string	0 — no immediate response string
monitor mode	disabled
hardwire handshake (pin 20)	enabled

Chapter 11
HP-IL Interfacing

What You'll Learn in This Chapter

This chapter is only for 7470 owners with an HP-IL interface. HP 7470s with Option 003 have an HP-IL interface.

In this chapter, you will find a brief overview of HP-IL, a list of the HP-IL capabilities implemented on the 7470, and examples of sending and receiving data using a variety of computers.

An Overview of HP-IL

In an HP-IL system, devices are connected to each other in a closed loop. All devices communicate by sending messages consisting of 11 bits each; these messages travel through the loop in one direction, one bit at a time. Only one message is traveling around the loop at a given time.

There are three categories that describe whether devices can send or receive messages: talkers, listeners, and controllers.

- Talkers are devices that send data over the interface; only one talker can be active at a given time. The controller designates the role of talker with commands that are dependent on the specific controller. The 7470 is capable of being a talker.
- Listeners are devices that receive data from a talker or commands from a controller; several listeners can be active simultaneously. As with talkers, listeners are designated by the controller. The 7470 is capable of being a listener.
- Controllers are in charge of all loop operations. For example, the controller assigns the roles of talker and listener, assigns addresses, and initiates data transfer between devices. There can be more than one controller in a loop, but only one can be active at any time, and only one can be the system controller. A controller is typically a portable computer or calculator. The 7470 does not have the ability to be a controller.

For more detailed information on HP-IL and how your computer sends commands and data through the interface, refer to your HP-IL and computer documentation. Typically, BASIC statements are used to send interface commands and messages.

HP-IL Implementation on the 7470

The HP-IL capability subsets for the 7470 are listed in the following table.

Function	Description	Implementation
R	Receiver.	Complete capability.
AH	Acceptor handshake.	Complete capability.
SH1	Source handshake.	Complete capability.
D	Driver.	Complete capability.
L1	Listener.	Basic listener.
L3	Listener.	Unaddress if addressed to talk (MTA).
LE0	Extended listener.	No capability.
T1	Talker.	Basic talker; send data.
T2	Talker.	Send status. (Returns a byte containing the status that is sent with the HP-GL output status command, OS. Does not reset bit number 3, the initialize flag, in the status byte.)
T3	Talker.	Send device ID. (Returns the string "HP7470A" *followed* by a carriage return and a line feed.)
T4	Talker.	Send accessory ID. (Returns a byte with the following bit pattern: 0110 0000 or 60 hex.)
T6	Talker.	Unaddress if addressed to listen (MLA).

Function	Description	Implementation
TE0	Extended talker.	No capability.
C0	Controller.	No capability.
DC0	Device clear.	No capability.
DT0	Device trigger.	No capability.
PP0	Parallel poll.	No capability.
SR0	Service request.	No capability.
AA1	Auto address.	Complete capability.
AE0	Auto extended address.	No capability.
AM0	Auto multiple address.	No capability.
RL0	Remote local.	No capability.
PD0	Power down.	No capability.
DD0	Device dependent commands.	No capability.

Reaction to Interface Commands and Messages

All unsupported interface commands are ignored and retransmitted around the loop. Upon receipt of the interface clear (IFC) message, the plotter resets the parser and starts looking for a new HP-GL instruction. Any partially parsed HP-GL instruction or parameters will be lost. This message does *not* reset parameters in the plotter to their default values.

Addressing the Plotter

The default address of the plotter is 5. However, the plotter address in a program may vary; this is because the system controller generally assigns addresses automatically in a sequential manner around the loop. Refer to your computer's documentation for hints on writing programs so that they will run no matter what order the peripheral devices are interconnected. The examples in this chapter assume that the plotter is the only device connected to the computer. Therefore, the plotter's address is 1.

Sending and Receiving Data

Computer-to-Plotter

Transmitting data from a computer to the plotter is typically accomplished using I/O statements such as PRINT, PRINT#, or OUTPUT. The following examples of sending program data to the plotter from various computers are only intended to illustrate the necessity for understanding the I/O statement protocol implemented by your computer. Each of these examples will cause the plotter to label the identity of the computer sending data, beginning at the X,Y coordinates 1000,2000. The examples involve sending both character string and numeric data as variables, and constants or literals.

HP-41 RPN Example:

NOTE: The characters that are enclosed in quotation marks must be entered in the alpha mode (the quotation marks do not need to be entered). The "├" symbol is the "alpha append" symbol; it is produced by pressing the shift and K keys while in the alpha mode. ■

```
01◆LBL "CTP"
02 AUTOIO
03 FIX 0
04 CF 29
05 2000
06 "SP1 PA1000,"
07 ARCL X
08 OUTA
09 41
10 "LBHP "
11 ARCL X
12 "├ SENDING DATA"
13 0
14 ENTER↑
15 3
16 BLDSPEC
17 ARCL X
18 OUTA
19 "SPO"
20 OUTA
21 END
```

Result: HP 41 SENDING DATA

HP-75 BASIC Example:

```
10 ASSIGN IO ":PL"
20 PRINTER IS ":PL"
30 A$="SENDING DATA"
40 B=75
50 Y=2000
60 PRINT "SP1;PA1000,",Y
70 PRINT "LBHP";B;A$;CHR$(3)
80 PRINT "SP0;"
90 END
```

Result: **HP 75 SENDING DATA**

HP Series 80 BASIC Example:

```
10 PRINTER IS 901
20 A$="SENDING DATA"
30 B=80
40 Y=2000
50 PRINT "SP1;PA1000,",Y
60 PRINT "LBHP";B;A$;CHR$(3)
70 PRINT "SP0;"
80 END
```

Result: **HP 80 SENDING DATA**

Plotter-to-Computer

Transmitting data from the plotter to the computer is typically accomplished using I/O statements such as READ, INPUT, and ENTER. Sometimes these statements are only available in I/O ROMs; check your computer's documentation or ask your HP dealer or HP Sales and Support Office. The following examples of obtaining output data from the plotter using various computers are only intended to illustrate the necessity for understanding the I/O statement protocol implemented on your computer. Each of these examples commands the pen to move to plotter coordinates 1000,1000 and then output the current pen position and the plotter identifier string to the computer.

HP-41 RPN Example:

```
01◆LBL "PTC"
02 AUTOIO
03 CF 21
04 "PA1000,1000 OC"
05 OUTA
06 INA
07 AVIEW
08 TONE 9
09 PSE
10 "OI"
11 OUTA
12 INA
13 AVIEW
14 END
```

HP-41 beeps and displays: 1000,1000,0

HP-41 display is then replaced with: 7470A

HP-75 BASIC Example:

At the time of this printing, the I/O functions necessary to obtain data *from* the plotter are not available for the HP-75. Please contact your dealer or HP Sales and Support Office for more information. (Data can be sent *to* the plotter without additional I/O functions. Refer to the example in the previous section.)

HP Series 80 BASIC Example:

```
10 PRINTER IS 901
20 PRINT "PA1000,1000;OC"
30 ENTER 901;A,B,C
40 PRINT "OI;"
50 ENTER 901;A$
60 DISP A,B,C,A$
70 END
```

Displayed current pen position and identification (HP-85):

```
1000                1000
0                   7470A
```

Displayed current pen position and identification (HP-86/87):

```
1000        1000        0        7470A
```

Appendix A
An HP-IB Overview

The HP Interface Bus (HP-IB) provides an interconnecting channel for data transfer between devices on the HP-IB.

The following list defines the terms and concepts used to describe HP-IB (bus) system operations.

HP-IB System Terms

1. **Addressing** — the characters sent by a controlling device specifying which device sends information on the bus and which device(s) receives the information.
2. **Byte** — a unit of information consisting of 8 binary digits (bits).
3. **Device** — any unit that is compatible with the ANSI/IEEE 488-1978 Standard.
4. **Device Dependent** — a response to information sent on the HP-IB that is characteristic of an individual device's design, and may vary from device to device.
5. **Operator** — the person that operates either the system or any device in the system.
6. **Polling** — the process typically used by a controller to locate a device that needs to interact with the controller. There are two types of polling:
   - **Serial Poll** — a method which obtains one byte of operational information about an individual device in the system. The process must be repeated for each device from which information is desired.
   - **Parallel Poll** — a method for obtaining information about a group of devices simultaneously.

Interface Bus Concepts

Devices which communicate along the interface bus can be classified into three basic categories.

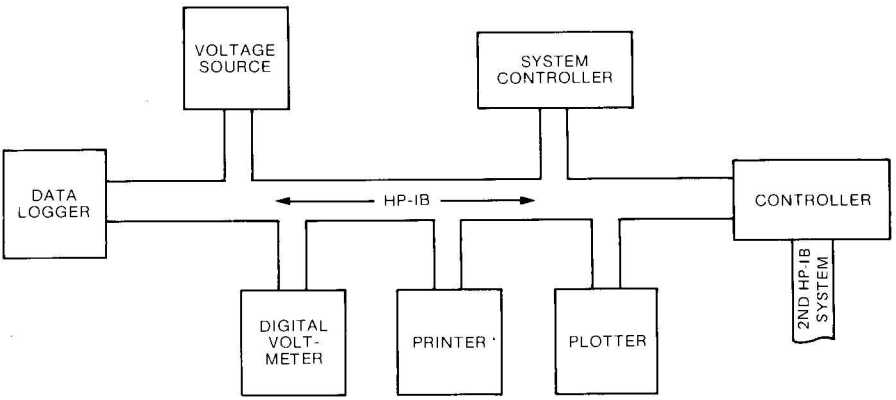
1. **Talkers** — devices which send information on the bus when they have been addressed.

2. **Listeners** — devices which receive information sent on the bus when they have been addressed.

3. **Controllers** — devices that can specify the talker and listeners for an information transfer. Controllers can be categorized as one of two types:

   - **Active Controller** — the current controlling device on the bus. Only one device can be the active controller at any time.
   - **System Controller** — the only controller that can take priority control of the bus if it is not the current active controller. Although each bus system can have only one system controller, the system can have any number of devices capable of being the active controller.

A typical HP-IB system is shown below.



Message Concepts

Devices which communicate along the interface bus are transferring quantities of information. The transfer of information can be from one device to another device, or from one device to more than one device. These quantities of information can easily be thought of as "messages."

In turn, the messages can be classified into 12 types. The list below gives the 12 message types for the HP-IB.

1. **The Data Message.** This is the actual information which is sent from one talker to one or more listeners along the interface bus.

2. **The Trigger Message.** This message causes the listening device(s) to perform a device-dependent action when addressed.

3. **The Clear Message.** This message causes either the listening device(s) or all of the devices on the bus to return to their predefined device-dependent states.

4. **The Remote Message.** This message causes all devices currently addressed to listen to switch from local front-panel control to remote program control.

5. **The Local Message.** This message clears the Remote Message from the listening device(s) and returns the device(s) to local front-panel control.

6. **The Local Lockout Message.** This message prevents a device operator from manually inhibiting remote program control.

7. **The Clear Lockout/Local Message.** This message causes all devices on the bus to be removed from Local Lockout and revert to Local. This message also clears the Remote Message for all devices on the bus.

8. **The Require Service Message.** A device can send this message at any time to signify that the device needs some type of interaction with the controller. This message is cleared by sending the device's Status Byte Message if the device no longer requires service.

9. **The Status Byte Message.** A byte that represents the status of a single device on the bus. Bit 6 indicates whether the device sent a Require Service Message, and the remaining bits indicate operational conditions defined by the device. This byte is sent from a talking device in response to a serial poll operation performed by a controller.

10. **The Status Bit Message.** This byte represents the operational conditions of a group of devices on the bus. Each device responds on a particular bit of the byte thus identifying a device-dependent condition. This bit is typically sent by devices in response to a parallel poll operation.

    The Status Bit Message can also be used by a controller to specify the particular bit and logic level at which a device will respond when a parallel poll operation is performed. Thus, more than one device can respond on the same bit.

11. **The Pass Control Message.** This transfers the bus management responsibilities from the active controller to another controller.

12. **The Abort Message.** The system controller sends this message to unconditionally assume control of the bus from the active controller. This message terminates all bus communications (but does not implement a Clear Message).

These messages represent the full implementation of all HP-IB system capabilities. Each device in a system may be designed to use only the messages that are applicable to its purpose in the system. It is

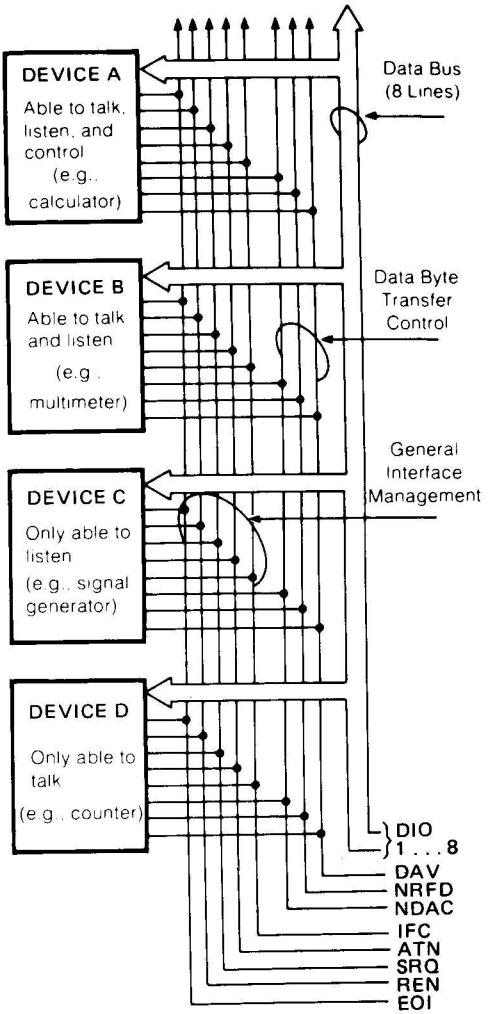
important for you to be aware of the HP-IB functions implemented on each device in your HP-IB system to ensure the operational compatibility of the system.

The HP Interface Bus

HP-IB Lines and Operations

The HP Interface Bus transfers data and commands between the components of an instrumentation system on 16 signal lines. The interface functions for each system component are performed within the component so only passive cabling is needed to connect the systems. The cables connect all instruments, controllers, and other components of the system in parallel to the signal lines.

The eight Data I/O lines (DIO1 through DIO8) are reserved for the transfer of data and other messages in a byte-serial, bit-parallel manner. Data and message 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 figure on the right.



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 or by jumpers on

a PC board. While the ATN line is true, all devices must listen to the data lines. When the 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 be automatically 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 on only seven of the data lines using the ASCII code. When ATN is false, any code of eight bits or less understood by both talker and listener(s) may be used.

The IFC (interface clear) line places the interface system in a known quiescent state.

The REN (remote enable) line is used with the Remote, Local, and Clear Lockout/Set Local messages to select either local or remote control of each device.

Any active device can set the SRQ (service request) line true via the Require Service Message. This indicates to the controller that some device on the bus wants attention, such as a counter that has just completed a time-interval measurement and wants to transmit the reading to a printer.

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 line assigned to it.

In the interest of cost-effectiveness, it is not necessary for every device to be capable of responding to all the lines. Each can be designed to respond only to those lines that are pertinent to its function on the bus.

The operation of the interface is generally controlled by one device equipped to act as controller. The interface transmits a group of commands to direct the other instruments on the bus in carrying out their functions of talking and listening.

The controller has two ways of sending interface messages. Multi-line messages, which cannot exist concurrently with other multi-line

messages, are sent over the eight data lines and the three handshake lines. Uni-line messages are transferred over the five individual lines of the management bus.

The commands serve several different purposes:

- Addresses or talk and listen commands select the instruments that will transmit and accept data. They are all multi-line messages.
- Unviersal commands cause every instrument equipped to do so to perform a specific interface operation. They include multi-line messages and three uni-line 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 multi-line commands. An instrument responds to an addressed command, however, only after an address has already told it to be talker or listener.
- Secondary commands are multi-line 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 an instrument, the controller uses seven of the eight data-bus lines. This allows instruments using the ASCII 7-bit code to act as controllers. As shown in the following table, 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.)

Command and Address Codes

Code Form									Meaning
X	0	0	A_5	A_4	A_3	A_2	A_1		Universal Commands
X	0	1	A_5	A_4	A_3	A_2	A_1		Listen Addresses
			except						
X	0	1	1	1	1	1	1		Unlisten Command
X	1	0	A_5	A_4	A_3	A_2	A_1		Talk Address
			except						
X	1	0	1	1	1	1	1		Untalk Command
X	1	1	A_5	A_4	A_3	A_2	A_1		Secondary Commands
			except						
X	1	1	1	1	1	1	1		Ignored

Code used when attention (ATN) is true (low).
X = don't care.

Interface Functions

Interface functions provide the physical capability to communicate via HP-IB. These functions are defined in the ANSI/IEEE 488-1978 Standard. This standard, which is the designer's guide to the bus, defines each interface function in terms of state diagrams that express all possible interactions.

Bus capability is grouped under 10 interface functions, for example: Talker, Listener, Controller, Remote/Local. The following table lists the functions, including two special cases of Controller.

HP-IB Interface Functions

Mnemonic	Interface Function Name
SH	Source Handshake
AH	Acceptor Handshake
T	Talker (or TE = Extended Talker)*
L	Listener (or LE = Extended Listener)*
SR	Service Request
RL	Remote Local
PP	Parallel Poll
DC	Device Clear
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.

Bus Messages

Since interface functions are the physical agency through which bus messages are implemented, each device must implement one or more functions to enable it to send or receive a given bus message.

The following table lists the functions required to implement each bus message. Each device's operating manual lists the functions implemented by that device. Some devices, such as the 98034A Interface, list the functions implemented directly on the device.

Functions Used by Each Bus Message

Bus Message	Functions Required sender function → receiver function(s) (support functions)
Data	T → L* (SH, AH)
Trigger	C → DT* (L, SH, AH)
Clear	C → DC* (L, SH, AH)
Remote	C_S → RL* (SH, AH)
Local	C → RL* (L, SH, AH)
Local Lockout	C → RL* (SH, AH)
Clear Lockout/Set Local	C_S → RL*
Require Service	SR* → C
Status Byte	T → L* (SH, AH)
Status Bit	PP* → C
Pass Control	C_A → C_B (T, SH, AH)
Abort	C_S → T, L*C

*Since more than one device can receive (or send) this message simultaneously, each device must have the function indicated by an * .

Appendix B
Instruction Syntax

HP-GL Syntax

This section lists the formal syntax for each plotter instruction in alphabetical order of the instruction's two-letter mnemonic.

Each instruction is listed with its purpose, syntax, parameter or response type, and range. If no parameter range is given, the range is -2^{15} to $2^{15}-1$. Refer to the indicated pages for details. The semicolon is included as the terminator for all instructions except the label instructions. A nonalphabetic or nonnumeric character such as # or $, or the next mnemonic can also be used as the instruction terminator. In addition, if you have an HP-IB or HP-IL plotter, the line feed character can be used as a terminator. The semicolon appears in parentheses (;) if the instruction executes without the plotter receiving the terminator. [TERM] means the terminator sent by the plotter at the end of output. It is CR LF in an HP-IB or HP-IL configuration and CR or as set by an ESC . M command in an RS-232-C configuration.

AA* The Arc Absolute Instruction

Page 3-17

AA X-coordinate,Y-coordinate,arc angle(,chord angle);

Purpose: Draws arc of specified number of degrees with specified smoothness; centered at X,Y coordinate, using current pen status (up or down).

Parameters: X- and Y-coordinates — integer, in plotter units unless scaling in effect; then in user units.

arc angle — integer, negative value specifies clockwise arc, positive value specifies counterclockwise arc.

chord angle — integer, defines arc smoothness in degrees. Default is 5 degrees.

*Available only with RS-232-C plotters that have the serial prefix number **2308A** or higher.

AR* The Arc Relative Instruction

Page 3-19

AR X-increment,Y-increment,arc angle(,chord angle);

Purpose: Draws arc of specified number of degrees with specified smoothness; centered relative to current pen position, using current pen status (up or down).

Parameters: X- and Y-increments — integer, in plotter units unless scaling in effect; then in user units.

arc angle — integer, negative value specifies clockwise arc, positive value specifies counterclockwise arc.

chord angle — integer, defines arc smoothness in degrees. Default is 5 degrees.

CA The Designate Alternative Character Set Instruction

Page 5-4

CA n(;)

Purpose: Designates the alternate character set.

Parameter: integer 0 through 4; default set 0.

CI* The Circle Instruction

Page 3-12

CI radius(,chord angle);

Purpose: Draws a circle of specified radius centered at current pen position.

Parameters: radius — integer, in plotter units unless scaling in effect; then in user units. Starting point at 0 degrees with positive parameter; 180 degrees with negative parameter.

chord angle — integer, defines circle smoothness in degrees. Default is 5 degrees.

CP The Character Plot Instruction

Page 5-13

CP spaces, lines;

Purpose: Move the pen the number of spaces and lines specified.

Parameters: spaces — decimal, ≥ -128 and < 128, number of CP spaces, positive value moves pen in current label direction, negative value moves pen in opposite direction.

lines — decimal, ≥ -128 and < 128, number of CP lines, positive value moves pen up, negative value moves pen down in relation to current label direction.

Omitting parameters causes carriage return, line feed.

*Available only with RS-232-C plotters that have the serial prefix number **2308A** or higher.

CS The Designate Standard Character Set Instruction

Page 5-3

CS m (;)

Purpose: Designates the standard character set.

Parameter: integer, 0 through 4; default set 0.

DC The Digitize Clear Instruction

Page 6-3

DC (;)

Purpose: Clears digitize mode without entering a point from the front panel.

DF The Default Instruction

Page 1-10

DF ;

Purpose: Returns plotter to default conditions. See the table in Appendix C.

DI The Absolute Direction Instruction

Page 5-10

DI run, rise ;

Purpose: Sets the direction of labels.

Parameters: run, rise — decimal values, unitless. At least one must be nonzero, i.e., | parameter | ≥ 0.0004 .

Omitting parameters causes horizontal labels and is the same as DI 1 , 0 .

DP The Digitize Point Instruction

Page 6-2

DP (;)

Purpose: Places plotter in digitize mode waiting for point to be entered from front panel.

DR The Relative Direction Instruction

Page 5-11

DR run, rise ;

Purpose: Sets the direction of labels.

Parameters: decimals, −128 to +127.9999 .

run is % of ($P2_x - P1_x$), rise is % of ($P2_y - P1y$).

Omitting parameters causes horizontal labels as does DR 1 , 0 .

DT The Define Terminator Instruction

Page 5-6

DT t(;)

Purpose: Defines the label terminator used in LB command.

Parameter: ASCII character 1 to 127 except 5 and 27. Only an IN or DF command or use of ETX (decimal 3) as parameter restores label terminator to ETX, its default value.

IM The Input Mask Instruction

Page 1-12

IM E-mask value (, S-mask value(, P-mask value)) (;)

Purpose: Set masks to specify which errors will cause the **ERROR** LED to come on and bit 5 of the status byte to be set, and to specify what conditions will cause a positive response to a serial or parallel poll in an HP-GL environment.

Parameters: integers 0 through 255. If parameters omitted, masks are set to 223,0,0, the default values.

IN The Initialize Instruction

Page 1-11

IN ;

Purpose: Sets the plotter to default conditions plus raises the pen, sets the scaling points to P1 = 250,279 and P2 = 10 250,7479, clears all HP-GL errors, sets bit 3 of the output status byte to true (1), and reads setting of paper switch.

IP The Input P1 and P2 Instruction

Page 2-4

IP $P1_x$, $P1_y$ (, $P2_x$, $P2_y$) (;)

Purpose: Sets scaling points.

Parameters: Integers in plotter units. Omitting parameters sets P1 and P2 to default values, P1 = 250,279, P2 = 10 250,7479.

IW The Input Window Instruction

Page 2-9

IW $X_{lower\ left}$, $Y_{lower\ left}$, $X_{upper\ right}$, $Y_{upper\ right}$ (;)

Purpose: Sets window inside which plotting can occur.

Parameters: Specify X- and Y-coordinates of lower-left and upper-right corners of the window.

Omitting parameters sets window to maximum plotting area, determined by the setting of the paper switch.

LB The Label Instruction

Page 5-7

LB c...c t

Purpose: Draws the character string using the currently selected character set.

Parameters: c...c — ASCII characters which may include control characters.

Terminator: t — label terminator defined by DT. Default is ETX, decimal 3.

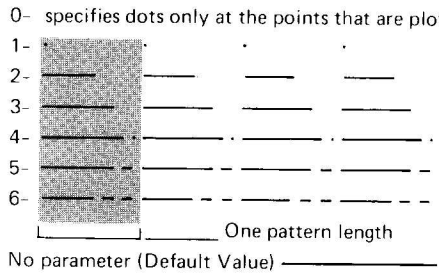
LT The Line Type Instruction

Page 4-6

LT pattern number (, pattern length) (;)

Purpose: Sets the line type used in drawing lines.

Parameters: pattern number — integer between 0 and +6. Omitting parameter causes solid line.



pattern length — decimal, 0 to 127.9999, a percentage of diagonal distance between P1 and P2. Default 4%.

OA The Output Actual Position and Pen Status Instruction

Page 7-3

OA (;)

Purpose: Used to output the pen's physical position at time of command.

Response: X,Y,P [TERM] — integers, in ASCII.

X,Y — in plotter units within current window.

P — 0, pen up or 1, pen down.

OC The Output Commanded Position and Pen Status Instruction

Page 7-4

OC (;)

Purpose: Used to output the pen position and status at time of command.

Response: X,Y,P [TERM] — decimal numbers,* in ASCII.

X,Y — −32 768 to 32 767.

P — 0, pen up or 1, pen down.

Plotter units unless scaling in effect; then in user units.

OD The Output Digitized Point and Pen Status Instruction

Page 6-3

OD (;)

Purpose: Used to output the physical pen position and status for the last digitized point.

Response: X,Y,P [TERM] — integers, in ASCII.

X,Y — In plotter units, within mechanical limits.

P — 0, pen up or 1, pen down.

OE The Output Error Instruction

Page 7-5

OE (;)

Purpose: Used to output the last HP-GL error.

Response: error number [TERM] — a positive ASCII integer, 0 through 8, excluding 4.

OF The Output Factors Instruction

Page 7-6

OF (;)

Response: 40, 40 [TERM] — integers, in ASCII.

OI The Output Identification Instruction

Page 7-7

OI (;)

Purpose: Used to output the plotter's identification.

Response: 7470A [TERM] — ASCII string.

*If you have an HP-IB or RS-232-C plotter that has a serial prefix number *lower than* **2308A**, OC parameters are output as integers. For more information, refer to the explanation of the OC instruction on page 7-4.

OO The Output Options Instruction

Page 7-7

OO (;)

Purpose: Used to output features implemented on the plotter.

Response: 0,1,0,0,1,0,0,0 [TERM]

- 1 (second): Indicates arcs and circle instructions are included (available only with RS-232-C plotters that have the Serial Prefix number **2308A** or higher).
- 1 (first): Indicates pen select capability is included (available on all plotters).

OP The Output P1 and P2 Instruction

Page 2-5

OP (;)

Purpose: Used to output the plotter unit coordinates of the scaling points P1 and P2.

Response: $P1_x$, $P1_y$, $P2_x$, $P2_y$ [TERM] — four integers in ASCII.

Range — dependent on settings of paper switch.

US	A4
$0 \leqslant$ X-coordinate $\leqslant$ 10 300	$0 \leqslant$ X-coordinate $\leqslant$ 10 900
$0 \leqslant$ Y-coordinate $\leqslant$ 7650	$0 \leqslant$ Y-coordinate $\leqslant$ 7650

OS The Output Status Instruction

Page 7-8

OS (;)

Purpose: Used to output the plotter's status.

Response: status [TERM] — integer in ASCII in the range 0 to 255. Power-on status, 24.

OW The Output Window Instruction

Page 2-10

OW (;)

Purpose: Used to output the plotter unit coordinates of the lower-left and upper-right corners of the current window.

Response: $X_{\text{lower left}}$, $Y_{\text{lower left}}$, $X_{\text{upper right}}$, $Y_{\text{upper right}}$ [TERM] — integers in ASCII. Range same as OP.

PA The Plot Absolute Instruction

Page 3-4

PA X_1 coordinate, Y_1 coordinate (X_2 coordinate, Y_2 coordinate, . . ., . . ., X_n coordinate, Y_n coordinate) (;)

or

PA (;)

Purpose: Plots to the X,Y coordinates in the order listed using the current pen up/down status. PA; sets absolute plotting.

Parameters: Pairs of integers representing plotter units if scaling not in effect, otherwise user units, integers or decimals.

PD The Pen Down Instruction

Page 3-2

PD (;)

or

PD X_1 coordinate, Y_1 coordinate (,... X_n, Y_n coordinates) (;)

Purpose: Programmatically lowers the pen. Parameters may be included as in PA or PR.

PR The Plot Relative Instruction

Page 3-8

PR X_1 increment, Y_1 increment (, X_2 increment, Y_2 increment, . . ., . . ., X_n increment, Y_n increment) (;)

or

PR (;)

Purpose: Plots, in order, to the points indicated by the X,Y increments, relative to the previous pen position. PR; sets relative plotting for PU or PD with parameters.

Parameters: Pairs of integers representing plotter units if scaling is not in effect, otherwise user units, integers or decimals.

PU The Pen Up Instruction

Page 3-2

PU (;)

or

PU X_1 coordinate, Y_1 coordinate (, . . . X_n, Y_n coordinates) (;)

Purpose: Programmatically raises the pen. Parameters may be included as in PA or PR.

SA The Select Alternate Character Set Instruction

Page 5-5

SA (;)

Purpose: Selects the alternate character set designated by the CA instruction as the character set to be used for subsequent labeling.

SC The Scale Instruction

Page 2-6

SC X_{min}, X_{max}, Y_{min}, Y_{max} (;)

Purpose: Scales the plotting area into user units.

Parameters: Integers.

SI The Absolute Character Size Instruction

Page 5-15

SI width, height ;

Purpose: Sets character width and height in centimetres for labels.

Parameters: width, height — decimals representing centimetres, −128 to +127.9999.

Omitting parameters establishes size of 0.19, 0.27, the same as the default SR sizing with default P1,P2.

SL The Character Slant Instruction

Page 5-18

SL tan θ (;)

Purpose: Establishes the slant for labeled characters.

Parameters: decimal, −128 to +127.9999, interpreted as the tangent of the angle from vertical.

Omitting parameters establishes no slant, the same as the default or SL 0.

SM The Symbol Mode Instruction

Page 4-4

SM character (;)

Purpose: Causes specified symbol to be drawn at each plotted point.

Parameter: Any printing character ASCII 33 through 127 excluding semicolon (ASCII 59). SM space, SM control character, or SM; cancels symbol mode.

SP The Pen Select Instruction

Page 3-2

SP pen number (;)

Purpose: Selects or stores a pen.

Parameter: integers. Omitting parameters or a parameter of 0 stores the pen. Odd-numbered parameter selects pen from left stall, even-numbered from right.

SR The Relative Character Size Instruction

Page 5-16

SR width, height ;

Purpose: Sets the character width and height relative to P1 and P2 for labels.

Parameters: decimals representing a percentage of vertical or horizontal distance between P1 and P2.

Width — percentage of $(P2_x - P1_x)$.

Height — percentage of $(P2_y - P1_y)$.

Omitting parameters results in value 0.75 for width and 1.5 for height.

SS The Select Standard Character Set Instruction

Page 5-4

SS (;)

Purpose: Selects the standard character set designated by the CS instruction as the character set used for subsequent labeling.

TL The Tick Length Instruction

Page 4-2

TL tp (, tn) (;)

Purpose: Establishes the length of ticks drawn with the instructions XT and YT.

Parameters: decimals.

tp — percentage of $(P2_y - P1_y)$ for XT or $(P2_x - P1_x)$ for YT. Denotes portion above the X-axis or to the right of the Y-axis when difference is positive.

tn — same as tp except denotes portion below the X-axis and to the left of the Y-axis.

Omitting parameters causes tick lengths tp and tn 0.5% of $(P2_y - P1_y)$ or $(P2_x - P1_x)$, the same as the default values.

UC* The User Defined Character Instruction

Page 5-19

UC (pen control ,) X-increment, Y-increment (,...) (, pen control) (,...) ;

Purpose: Draws characters or symbols defined by user.

Parameters: pen control — $\geq +99$ pen down or ≤ -99 pen up.

X-increment, Y-increment in grid units, range, ± 98 grid units.

Omitting parameters causes the pen to move one character-space field to the right.

VS The Velocity Select Instruction

Page 3-3

VS pen velocity (;)

Purpose: Sets the pen velocity.

Parameters: decimal, 0 to 127.9999.

pen velocity — 1 through 38.1 interpreted as cm/s. Defaults to velocity of 38.1 cm/s, acceleration of 2 g. Any velocity parameter slows acceleration to 0.5 g.

XT The X-Tick Instruction

Page 4-2

XT (;)

Purpose: Draws a vertical tick mark of the length specified by the TL instruction at the current pen position.

YT The Y-Tick Instruction

Page 4-2

YT (;)

Purpose: Draws a horizontal tick mark of the length specified by the TL instruction at the current pen position.

*Not available with Option 003.

RS-232-C Instruction Syntax

This section lists the formal syntax for each RS-232-C device control instruction in alphabetical order of the escape sequence. Refer to the indicated page for details.

Plotter On

Page 10-24

ESC . (or ESC . Y

Purpose: Places the plotter in a programmed-on state.

Plotter Off

Page 10-24

ESC .) or ESC . Z

Purpose: Places the plotter in a programmed-off state.

Set Plotter Configuration

Page 10-25

ESC . @ [(<DEC>) ; (<ASC>)] :

Purpose: Enables or disables hardwire handshake mode.

Parameters: <DEC> — Ignored.

<ASC> — Data Terminal Ready (CD) line control. ASCII decimal equivalent of 4-bit word (0 to 15).

Output Buffer Space

Page 10-26

ESC . B

Purpose: Outputs the number of byte spaces currently available for data in the buffer.

Response: <DEC> [TERM] — 0 to 255.

Output Extended Error

Page 10-27

ESC . E

Purpose: Outputs a decimal code to identify the type of RS-232-C related error that occurred.

Response: <DEC> [TERM] — 0, no error, or 10 - 16.

Set Handshake Mode 1

Page 10-28

ESC . H [(<DEC>) ; (<ASC>) ; (<ASC>(; . . . <ASC>))] :

Purpose: Establishes parameters for handshake mode 1, used when response to enquiry character requires ESC . M parameters.

Parameters: <DEC> — Block size or Xoff threshold level.

<ASC> — Enquiry character or not used.

<ASC> . . . <ASC> — Acknowledgment string of 1 to 10 characters or Xon trigger characters.

Set Handshake Mode 2

Page 10-29

ESC . I [(<DEC>) ; (<ASC>) ; (<ASC>(; . . . <ASC>))] :

Purpose: Establishes parameters for handshake mode 2, used when response to enquiry character does not require ESC . M parameters.

Parameters: <DEC> — Block size or Xoff threshold level.

<ASC> — Enquiry character or omitted.

<ASC> . . . <ASC> — Acknowledgment string of 1 to 10 characters or Xon trigger characters.

Abort Device Control

Page 10-31

ESC . J

Purpose: Aborts any partially decoded or executed device control instructions including outputs.

Abort Graphic Instruction

Page 10-32

ESC . K

Purpose: Aborts any partially decoded HP-GL instruction and discards instructions in buffer.

Output Buffer Size

Page 10-32

ESC . L

Purpose: Outputs the buffer size.

Response: 255. Not output until the buffer is empty.

Set Output Mode

Page 10-33

ESC . M [(<DEC>) ; (<ASC>) ; (<ASC>) ; (<ASC>(; (<ASC>)) ; (<ASC>)] :

Purpose: Sets parameters for output.

Parameters: <DEC> — Turnaround delay, 0-54 612.

<ASC> — Output trigger character, ASCII 0-127.

<ASC> — Echo terminator character, ASCII 0-127.

<ASC> . . . <ASC> — 1 or 2 output terminators, ASCII 0-127, 0 terminates string.

<ASC> — Output initiator character, ASCII 0-127.

Set Extended Output and Handshake Mode

Page 10-34

ESC . N [(<DEC>) ; (<ASC>(; . . . <ASC>))] :

Purpose: Establishes extended parameters for any output command.

Parameters: <DEC> — Delay between output characters, 0-54 612.

<ASC> . . . <ASC> — Immediate response string of 1 to 10 characters. ASCII 0-127, 0 terminates string; or Xoff trigger characters.

Output Extended Status

Page 10-38

ESC . O

Purpose: Outputs the decimal equivalent value of a 16-bit immediate status word.

Response: <DEC> [TERM] — a value 40 or less.

Reset Handshake

Page 10-40

ESC . R

Purpose: Resets the handshake to its default value. It is the same as sending the commands ESC . @ , ESC . H , ESC . I , ESC . M , and ESC . N without parameters.

Appendix C
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, the positional values of the 1's are added up. From the above example, this would be:

$$2^5 + 2^3 + 2^0 = 32 + 8 + 1 = 41$$

To convert from decimal to binary, the decimal number is divided by 2. The remainder is the binary equivalent. For example:

		Remainder (read up)	
2 ⟌41	→	1	
2 ⟌20	→	0	
2 ⟌10	→	0	= Binary 101001
2 ⟌ 5	→	1	
2 ⟌ 2	→	0	
2 ⟌ 1	→	1	

Scaling Without Using the SC Instruction

The 7470 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 command, it may be convenient for you to write programs where numbers to be plotted are in some units other than plotter units. These "user units" can be converted into plotter units by the computer using the following equations:

$$X_{scaled} = \left[\frac{P2_x - P1_x}{U2_x - U1_x}\right] A_x + P1_x - U1_x \left[\frac{P2_x - P1_x}{U2_x - U1_x}\right]$$

$$Y_{scaled} = \left[\frac{P2_y - P1_y}{U2_y - U1_y}\right] A_y + P1_y - U1_y \left[\frac{P2_y - P1_y}{U2_y - U1_y}\right]$$

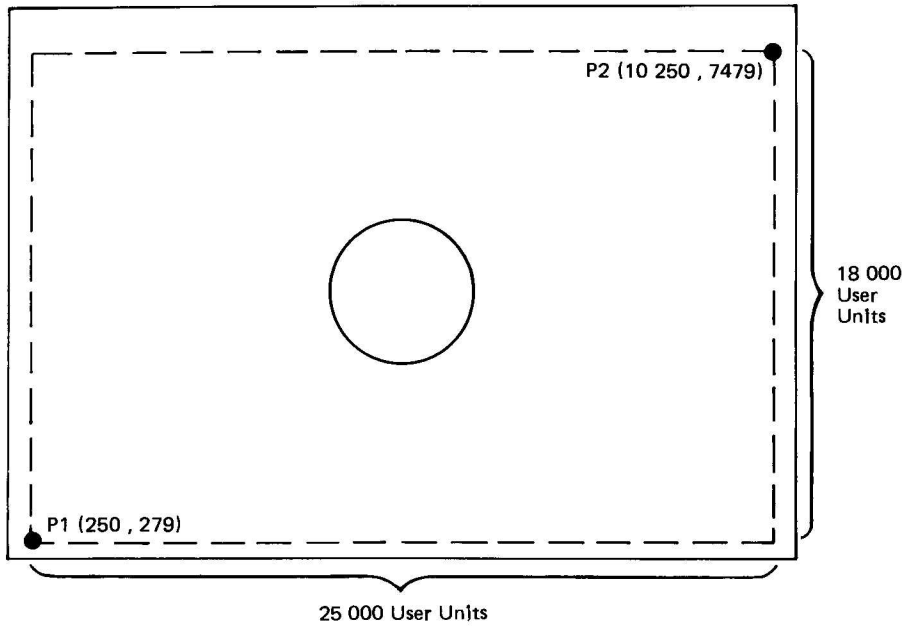
where: A_x is the X-coordinate of the desired point in user units,
A_y is the Y-coordinate of the desired point in user units,
$P1_x$ is the X-coordinate of P1 in plotter units,
$P1_y$ is the Y-coordinate of P1 in plotter units,
$P2_x$ is the X-coordinate of P2 in plotter units,
$P2_y$ is the Y-coordinate 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.

Example 1:

Problem

Scale the platen area (P1 = 250,279 and P2 = 10 250,7479) into user units where P1 = 0,0 and P2 = 25 000,18 000. At the center point (X = 12 500, Y = 9000), draw a circle with radius 2500 as follows:



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:

$A_x = 2500 \cos t + 12\,500$

$A_y = 2500 \sin t + 9000$

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 were set using DF or IN commands:

$P1 = 250, 279$

$P2 = 10\,250, 7479$

therefore

$P1_x = 250$

$P1_y = 279$

$P2_x = 10\,250$

$P2_y = 7479$

F. Solving for X and Y:

$$X = \left[\frac{P2_x - P1_x}{U2_x - U1_x}\right] A_x + P1_x - U1_x \left[\frac{P2_x - P1_x}{U2_x - U1_x}\right]$$

$$+ \left[\frac{10\,250 - 250}{25\,000 - 0}\right] 2500 \cos t + 12\,500 + 250 - 0 \left[\frac{10\,250 - 250}{25\,000 - 0}\right]$$

$$= 0.4\,(2500 \cos t + 12\,500) + 250 - 0$$

$$= 1000 \cos t + 5250$$

$$Y = \left[\frac{P2_y - P1_y}{U2_y - U1_y}\right] A_y + P1_y - U1_y \left[\frac{P2_y - P1_y}{U2_y - U1_y}\right]$$

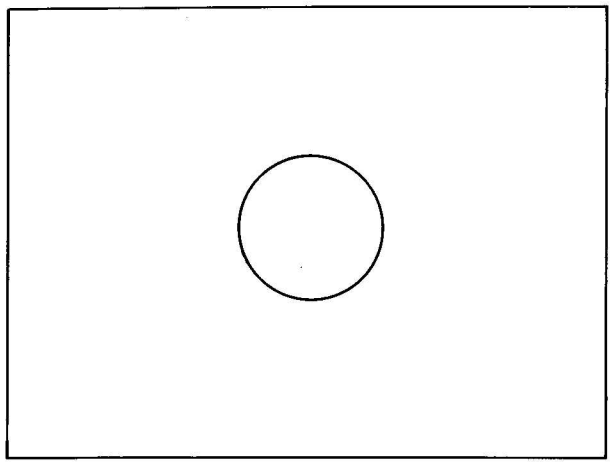
$$+ \left[\frac{7479 - 279}{18\,000 - 0}\right] 2500 \sin t + 9000 + 279 - 0 \left[\frac{7479 - 279}{18\,000 - 0}\right]$$

$$= 0.4\,(2500 \sin t + 9000) + 279 - 0$$

$$= 1000 \sin t + 3879$$

G. Sending the following program will plot the required circle using the default P1 and P2.

```
10   PRINT "IP250,279,10250,7479;SP1"
20   FOR T=0 TO 2*PI STEP PI/20
30   X=1000*COS(T)+5250
40   Y=1000*SIN(T)+3879
50   PRINT "PA";X;Y;"PD"
60   NEXT T
70   PRINT "SPO"
```



Plotter Default Conditions

Plotting mode	Absolute (PA)
Relative character direction	Horizontal (DR 1 , 0)
Line type	Solid line
Line pattern length	4% of the distance from P1 to P2
Input window	Mechanical limits of plotter
Relative character size	(SR .75 , 1.5) width = 0.75% of ($P2_x$ − $P1_x$) height = 1.5% of ($P2_y$ − $P1_y$)
Scale	Off
Symbol mode	Off
Tick length (on either side of axis)	0.5% of ($P2_x$ − $P1_x$) or ($P2_y$ − $P1_y$)
Standard character set	Set 0
Alternate character set	Set 0
Label terminator	ETX (ASCII decimal equivalent 3)
Character slant	0°
Mask value	223 , 0 , 0
Digitize clear	On
Pen velocity	38.1 cm/s (15 in./s)
*Chord angle	Set to 5 degrees for AA, AR, and CI

P1 and P2 are changed only with the initialize command (IN). They are not affected by device clear and the default command (DF).

*Applicable only to RS-232-C plotters that have the serial prefix number **2312A** or higher.

HP-GL Error Messages

error 0 No error.

error 1 Instruction not recognized. The plotter has received an illegal character sequence.

error 2 Wrong number of parameters. Too many or too few parameters have been sent with an instruction.

error 3 Bad parameter. The parameters sent to the plotter with an instruction are out of range for that instruction.

error 4 Not used.

error 5 Unknown character set. A character set out of the range 0 through 4 has been designated as either the standard or alternate character set.

error 6 Position overflow. An attempt to draw a character (LB or UC) or perform a CP that is located outside the plotter's numeric limit of −32 768 to +32 767.

error 7 Not used.

error 8 Vector received while pinch wheels raised.

RS-232-C Error Messages

0 No I/O error has occurred.

10 Output instruction received while another output instruction is executing. The original instruction will continue normally; the one in error will be ignored.

11 Invalid byte received after first two characters, ESC . , in a device control instruction.

12 Invalid byte received while parsing a device control instruction. The parameter containing the invalid byte 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 exit) or the first byte of another instruction is received (abnormal exit).

15 A framing error, parity error, or overrun error has been detected.

16 The input buffer has overflowed. As a result, one or more bytes of data have been lost, and therefore, an HP-GL error will probably occur.

The No Operation Instructions, NOP

In order to maintain software compatibility with the 9872 plotter, the 7470 recognizes six 9872-related instructions as no operation NOP instructions. These six NOP instructions are:

Automatic Pen Pickup AP	Advance Full Page AF
Adaptive Velocity VA	Advance Half Page AH
Normal Velocity VN	Enable Cutter EC

If these instructions are included in a program, they are recognized by the 7470 and implemented as a NOP (i.e., they are ignored).

On a 7470 plotter with an HP-IL interface, UC is also a NOP instruction.

ASCII Character Codes

Binary is often used as a code to represent not only numbers, but also alphanumeric characters such as "A" or "," or "x" or "2". One of the most common binary codes used is ASCII[1]. 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

A complete list of ASCII characters and their decimal representation and the characters drawn by the plotter in each of the five character sets are shown on the following pages. The five character sets are:

Set No.	Description
Set 0	ANSI ASCII
Set 1	9825 Character Set
Set 2	French/German
Set 3	Scandinavian
Set 4	Spanish/Latin American

[1]American Standard Code for Information Interchange.

7470 ASCII Code Definitions

Decimal Value	ASCII Character	All Sets
0	NULL	No Operation (NOP)
1	SOH	NOP
2	STX	NOP
3	ETX	End Label Instruction
4	ETO	NOP
5	ENQ	NOP
6	ACK	NOP
7	BEL	NOP
8	BS	Backspace
9	HT	NOP
10	LF	Line Feed
11	VT	Inverse Line Feed
12	FF	NOP
13	CR	Carriage Return
14	SO	Select Alternate Character Set
15	SI	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

NOTE: Characters offset to the left have the automatic backspace feature. ■

Decimal Value	Set 0	Set 1	Set 2	Set 3	Set 4
33	!	!	!	!	!
34	"	"	"	"	"
35	#	#	£	£	¿
36	$	$	$	$	$
37	%	%	%	%	%
38	&	&	&	&	&
39	'	'	´	'	´
40	(	(	(	(	(
41	)	)	)	)	)
42	*	*	*	*	*
43	+	+	+	+	+
44	,	,	,	,	,
45	-	-	-	-	-
46	.	.	.	.	.
47	/	/	/	/	/
48	0	0	0	0	0
49	1	1	1	1	1
50	2	2	2	2	2
51	3	3	3	3	3
52	4	4	4	4	4
53	5	5	5	5	5
54	6	6	6	6	6
55	7	7	7	7	7
56	8	8	8	8	8
57	9	9	9	9	9
58	:	:	:	:	:
59	;	;	;	;	;
60	<	<	<	<	<
61	=	=	=	=	=
62	>	>	>	>	>
63	?	?	?	?	?
64	@	@	@	@	@
65	A	A	A	A	A
66	B	B	B	B	B
67	C	C	C	C	C
68	D	D	D	D	D
69	E	E	E	E	E
70	F	F	F	F	F
71	G	G	G	G	G
72	H	H	H	H	H
73	I	I	I	I	I
74	J	J	J	J	J
75	K	K	K	K	K
76	L	L	L	L	L
77	M	M	M	M	M
78	N	N	N	N	N
79	O	O	O	O	O
80	P	P	P	P	P

7470 ASCII Code Definitions (Continued)

Decimal Value	Set 0	Set 1	Set 2	Set 3	Set 4
81	Q	Q	Q	Q	Q
82	R	R	R	R	R
83	S	S	S	S	S
84	T	T	T	T	T
85	U	U	U	U	U
86	V	V	V	V	V
87	W	W	W	W	W
88	X	X	X	X	X
89	Y	Y	Y	Y	Y
90	Z	Z	Z	Z	Z
91	[	[	[	Ø	[
92	\	f	ç	Æ	i
93	]	]	]	ø	]
94	^	↑	^	æ	^
95	_	_	_	_	_
96	`	`	`	`	`
97	a	a	a	a	a
98	b	b	b	b	b
99	c	c	c	c	c
100	d	d	d	d	d
101	e	e	e	e	e
102	f	f	f	f	f
103	g	g	g	g	g
104	h	h	h	h	h
105	i	i	i	i	i
106	j	j	j	j	j
107	k	k	k	k	k
108	l	l	l	l	l
109	m	m	m	m	m
110	n	n	n	n	n
111	o	o	o	o	o
112	p	p	p	p	p
113	q	q	q	q	q
114	r	r	r	r	r
115	s	s	s	s	s
116	t	t	t	t	t
117	u	u	u	u	u
118	v	v	v	v	v
119	w	w	w	w	w
120	x	x	x	x	x
121	y	y	y	y	y
122	z	z	z	z	z
123	{	π	¨	¨	~
124	¦	⊢	•	•	~
125	}	→	¨	¨	~
126	~	~	'	•	~
127	⊢	⊢	⊢	⊢	⊢

Subject Index

a

AA Instruction ..... 3-17 thru 3-19, B-1
AR Instruction ..... 3-19 thru 3-21, B-2
Absolute Direction Instruction, DI 5-10, 5-22, 5-23, B-3
Absolute Plotting 3-1, 3-4
Absolute Size Instruction, SI 5-13, 5-15, 5-22, B-9
Acceleration 1-5, 3-3
Acknowledgment String ..... 10-16, 10-21, 10-28 thru 10-31, 10-37
Addressing the Plotter, HP-IB 9-2, 9-3, 9-6
Addressing the Plotter, HP-IL 11-3
Arc Absolute Instruction, AA ..... 3-17 thru 3-19, B-1
Arc Relative Instruction, AR ..... 3-19 thru 3-21, B-2
ASCII Character Codes ..... C-7 thru C-10

b

Bar Graphs ..... 1-15, 8-10 thru 8-13
Baud Rate 10-12
Binary Coding and Conversions C-1
Binary-Decimal Conversions C-1
Block Size 10-16, 10-20, 10-21, 10-28, 10-29, 10-30, 10-37
Break Signal 10-6 10-7, 10-8, 10-24, 10-25
Buffer Space ..... 10-13 10-14, 10-16 thru 10-22, 10-26, 10-32, 10-38
Bus Commands 9-4

c

CA Instruction 5-4, 5-5, 5-7, B-2
CI Instruction ..... 3-12 thru 3-16, B-2
CP Instruction 5-13, 5-14, 5-15, 5-21, 8-3, 8-4, B-2
CS Instruction 5-3, 5-4, 5-7, B-3
Carriage-return Point 5-8, 5-10, 5-11, 5-14
CCITT V.24 Interface 1-1, 1-2, 10-1, 10-2, 10-10, 10-11, 10-12
Character Grid 5-19
Character Plot Instruction, CP 5-13, 5-14, 5-15, B-2
Character Sets ..... 5-2, C-7 thru C-10
Character Size 5-16, 5-23, B-13
Character Space Field 5-1, 5-13, 5-19
Circle Instruction, CI ..... 3-12 thru 3-16, B-2
Clipping 2-1, 2-9
Command Syntax, HP-GL 1-6
Command Syntax, RS-232-C 10-23
Connecting the RS-232-C Interface 10-10
Connector Cable, RS-232-C 10-10, 10-11, 10-12
Current Pen Position 3-1, 3-8, 5-7

Subject Index (Continued)

d

DC Instruction 6-3, B-3
DCL 9-4
DF Instruction 1-10, 3-4, 3-6, B-3
DI Instruction 5-10, 5-22, 5-23, B-3
DP Instruction ..... 6-2 thru 6-5, B-3
DR Instruction 5-11, 5-22, 5-25, B-3
DT Instruction 5-6, B-4
Data Block Size 10-16, 10-20, 10-21, 10-28, 10-29
Data Terminal Ready Line Control 10-16, 10-22, 10-25
Decimal Format 1-8
Default Conditions 1-11, C-5
Default Instruction, DF 1-10, 3-4, 3-6, B-3
Define Terminator Instruction, DT 5-6, B-4
Designate Alternate Character Set, CA 5-4, 5-5, 5-7, B-2
Designate Standard Set Instruction, CS 5-3, 5-4, 5-7, B-3
Device Clear 9-4
Device Control Instructions, RS-232-C ..... 10-1, 10-2, 10-3, 10-14, 10-22 thru 10-40, B-12, B-13, B-14
Digitize Clear Instruction, DC 6-3, B-3
Digitize Point Instruction, DP ..... 6-2 thru 6-5, B-3
Digitizing 6-1, 6-2, 6-4
Digitizing Sight 6-2
Documentation for the 7470 1-2

e

ESC . (..... 10-7, 10-10, 10-24, B-12
ESC .) 10-24, B-12
ESC . @ 10-8, 10-22, 10-25, 10-40, B-12
ESC . B 10-18, 10-26, B-12
ESC . E 10-13, 10-27, B-12
ESC . H 10-20, 10-21, 10-27, 10-28, 10-40, B-12
ESC . I 10-20, 10-21, 10-28, 10-29, 10-37, 10-38, 10-40, B-13
ESC . J 10-31, 10-32, B-13
ESC . K 10-32, B-13
ESC . L 10-32, B-13
ESC . M 10-17, 10-20, 10-23, 10-28, 10-33, 10-40, B-14
ESC . N 10-17, 10-20, 10-28, 10-34, 10-37, 10-40, B-14
ESC . O 10-38, B-14
ESC . R 10-40, B-14
ESC . Y 10-7, 10-10, 10-24, B-12
ESC . Z 10-24, B-12

Subject Index (Continued)

E-mask 1-12, 1-13
Eavesdrop Environment, RS-232-C 10-4
Echo Terminate Character 10-15, 10-17, 10-20, 10-21, 10-29, 10-31, 10-33, 10-36
Endline Environment 10-3
Enquire/Acknowledge Handshake ..... 10-14, 10-20, 10-21, 10-22, 10-28 thru 10-31, 10-39
Enquiry Character ..... 10-16, 10-20, 10-21, 10-28 thru 10-31, 10-37
Error Light 7-6, 10-13, 10-27
Error Messages, HP-IB 7-5, 7-6, C-6
Error Messages, RS-232-C 10-27, 10-28, C-6
ETX, End of Text Character 5-6, 5-11
Extended Status 10-38, 10-39
External Clock 10-12, 10-13

h

HP-GL Instruction Set ..... 1-6, 1-8, B-1 thru B-11
HP-GL Syntax ..... 1-6, 1-7, 1-8, B-1 thru B-11
HP-GL Error Status 1-12, 7-5, C-6
HP-IB ..... 7-6, A-1 thru A-8
HP-IB Implementation 9-2, A-2, A-3
HP-IB Interfacing ..... 9-1 thru 9-6, A-1 thru A-8
HP-IL Implementation 11-2, 11-3
HP-IL Interfacing ..... 11-1 thru 11-6
HP-IL Plotter Output 7-2
Half Duplex 10-10
Handshake Mode 1 ..... 10-28, 10-29, 10-30
Handshake Mode 2 ..... 10-28, 10-29, 10-30
Handshaking ..... 10-14 thru 10-22
Hard-clip Area 2-9
Hardwire Handshake 10-14, 10-22, 10-25
Hewlett-Packard Interface Loop ..... 1-2, 11-1 thru 11-6
Hewlett-Packard Interface Bus ..... 1-1, 9-2, A-1 thru A-8
Hewlett-Packard Graphics Language 1-1, 1-6

i

IFC 9-4
IM Instruction 1-12, 6-7, B-4
IN Instruction 1-11, 3-4, 3-5, 8-2, B-4
IP Instruction 2-4
IW Instruction 2-9, 8-13, B-4

Subject Index (Continued)

Immediate Response String 10-15, 10-20, 10-34
Initialize Instruction, IN 1-11, 3-4, 3-5, 8-2, 8-3
Input Mask Instruction, IM 1-12, 6-7, B-4
Input P1 and P2 Instruction, IP 2-4
Input Window Instruction, IW 2-9, 8-13, B-4
Integer Format 1-7
Intercharacter Delay 10-15, 10-17, 10-20, 10-29, 10-34, 10-36, 10-37
Interface Bus Concepts A-1
Interface Clear 9-4

l

LB Instruction 5-7, 8-4, B-5
LT Instruction 4-6, 8-6, 8-7, B-5
Label Fields 1-7, 5-7
Label Instruction, LB 5-7, 8-4, B-5
Label Terminator 5-6, 5-7
Labeling with Variables 5-8, 5-9
Line Graphs ..... 1-15, 8-1 thru 8-8
Line Type Instruction, LT 4-6, 8-6, 8-7, B-5
Listener 9-6

m

Mode 1 ..... 10-28, 10-29, 10-31
Mode 2 ..... 10-28, 10-29
Modem 10-4
Monitor Mode 10-8, 10-11, 10-25, 10-26

n

NOP, No Operation Instruction C-7

o

OA Instruction 7-3, 8-11, B-5
OC Instruction 7-4, B-6
OD Instruction ..... 6-3 thru 6-7, B-6
OE Instruction 7-5, B-6
OF Instruction 7-6, B-6
OI Instruction 7-7, B-6
OO Instruction 7-7, B-7
OP Instruction 2-5, B-7
OS Instruction 6-5, 7-8, B-7
OW Instruction 2-10, B-7

Subject Index (Continued)

On-line, Programmed Off State 10-6
On-line, Programmed On State 10-3, 10-7, 10-24
Optional Parameters 1-7, 1-8, 10-23
Output Actual Position
and Pen Status Instruction, OA 7-3, 8-11, B-5
Output Buffer Space Instruction, ESC . B 10-26, B-12
Output Commanded Position
and Pen Status Instruction, OC 7-4, B-6
Output Digitized Point
and Pen Status Instruction, OD ..... 6-3 thru 6-7, B-6
Output Error Instruction, OE 7-5, B-6
Output Extended Error Instruction, ESC . E 10-27, B-12, C-6
Output Extended Status Instruction, ESC . O 10-38, B-14
Output Factors Instruction, OF 7-6, B-6
Output Identification Instruction, OI 7-7, B-6
Output Initiator Character 10-15, 10-17, 10-20, 10-29, 10-34
Output Options Instruction, OO 7-7, B-7
Output P1 and P2 Instruction, OP 2-5
Output Status Instruction, OS 6-5, 7-8, B-7
Output Terminator 7-1, 7-2, 10-15, 10-17,
10-20, 10-21, 10-29, 10-33, 10-34, 10-37
Output Trigger Character 10-14, 10-17, 10-20, 10-21, 10-29, 10-36
Output Window Instruction, OW 2-10, 2-11, B-7

p

PA Instruction ..... 3-1, 3-4 thru 3-8, 8-7, B-8
PD Instruction ..... 3-2, 3-4, 3-6 thru 3-8, 8-7, B-8
PR Instruction 3-1, 3-8, 3-9, 3-10, B-8
PU Instruction ..... 3-2, 3-4 thru 3-8, 8-7, B-8
P-mask 1-12, 1-14, 6-7, 9-5
P1,P2 ..... 2-3 thru 2-8, 5-11, 5-16, 5-22 thru 5-26, 8-2
Paper Switch 2-2, 2-6, 2-10, 7-3
Parallel Poll 1-12, 1-14, 6-7, 9-5, A-1
Parameter Interaction in Labeling Commands ..... 5-21 thru 5-26
Pattern Number 4-6, B-5
Pen Down 3-2, 3-3, 3-4, 3-5, 3-8, 5-19, 5-20, B-8
Pen Instructions, PU and PD ..... 3-2, 3-4 thru 3-8, 8-7, B-8
Pen Up 3-2, 3-8, 5-19, 5-20, B-8
Pen Velocity 1-5, 3-3
Personal Computer 10-2, 10-3
Pie Charts 1-15, 8-10, 8-13
Pin Allocations, RS-232-C 10-11, 10-12
Plot Absolute Instruction, PA ..... 3-1, 3-4 thru 3-8, 8-7, B-8
Plot Relative Instruction, PR 3-1, 3-8, 3-9, 3-10, B-8

Subject Index (Continued)

Plotter Address 9-2, 9-3, 9-6
Plotter Character Sets ..... 5-2, C-7 thru C-10
Plotter Environments, RS-232-C ..... 10-2 thru 10-10
Plotter Instruction Set 1-6, 1-8
Plotter Off Instruction, ESC .) 10-24, B-12
Plotter On Instruction, ESC . (..... 10-24, B-12
Plotter Output 7-1
Plotter Syntax, 9872 ..... 1-7, 3-11
Plotter Unit 2-3, 7-2
Plotter Unit Equivalent 3-1, 3-9
Plotting Area 2-2
Plotting with Variables 3-8, 3-11, 8-7
Preparing Your Plotter for Digitizing 6-2

r

RS-232-C Interface 1-1, 10-1, 10-10, 10-11, 10-12
RS-232-C Interfacing ..... 10-1 thru 10-12
RS-232-C Plotter Output 7-2
Receiving Data, HP-IB 9-10
Receiving Data, HP-IL 11-5
Relative Direction Instruction, DR 5-11, 5-22, 5-25, B-3
Relative Plotting 3-1, 3-8, 3-9
Relative Size Instruction, SR ..... 5-16, 5-23 thru 5-26, B-10
Reset Handshake Instruction, ESC . R 10-40

s

SA Instruction 5-4, 5-5, 5-7, B-9
SC Instruction 2-6, 8-2, B-9
SDC 9-4
SI Instruction 5-15, 5-22, B-9
SL Instruction 5-18, B-9
SM Instruction 4-4, 5-27, B-9
SP Instruction 3-2, 8-2, B-10
SR Instruction ..... 5-16, 5-23 thru 5-26, B-10
SS Instruction 5-3, 5-4, 5-7, B-10
S-mask 1-12, 1-13, 6-7
Scale Instruction, SC 2-6, 8-2, B-9
Scaling 2-1, 2-6, 2-7, 2-8, 8-2, C-2
Scaling Points ..... 2-1, 2-3 thru 2-8, 5-11, 5-16, 5-22 thru 5-26, 8-2
Scaling Without Using the SC Instruction 2-7, C-2
Select Alternate Set Instruction, SA 5-4, 5-5, 5-7, B-9
Select Pen Instruction, SP 3-2, 8-2, B-10
Select Standard Set Instruction, SS 5-3, 5-4, 5-7, B-10

Subject Index (Continued)

Selective Device Clear 9-4
Sending Data, HP-IB 9-7
Sending Data, HP-IL 11-4
Serial Poll 6-7, 9-4
Service Request 6-7, 1-12
Set Handshake Mode 1 Instruction, ESC . H 10-20, 10-21, 10-27, 10-28, 10-40, B-13
Set Handshake Mode 2 Instruction, ESC . I 10-20, 10-21, 10-28, 10-29, 10-37, 10-38, 10-40, B-13
Set Plotter Configuration Instruction, ESC . @ 10-8, 10-22, 10-25, 10-40, B-12
Setting the Scaling Points 2-3
  Manually 2-4
  Programmatically 2-4, 8-2
Setting Up the Plotter, RS-232-C 10-2
Shift-in 5-3, 5-4
Shift-out 5-4, 5-5
Slant Instruction, SL 5-18, B-9
Software Checking Handshake 10-14, 10-17, 10-25
Spacing Between Characters 5-7, 5-13
Stand-alone Environment 10-3
Standard Character Set 5-2, 5-3, 5-4, 5-7
Stop Bits 10-13
Symbol Mode Instruction, SM 4-4, 4-5, 4-6, B-9

t

TL Instruction 4-2, B-10
Talker 9-6
Terminal ..... 10-4 thru 10-10
Terminal-only Environment 10-9, 10-10
Terminator ..... 1-6 thru 1-8
Tick Instructions, XT and YT 4-2, B-11
Tick Length Instruction, TL 4-2, 8-3, B-10
Tick Marks 4-2, 8-3
Transmission Errors, RS-232-C 10-13
Turnaround Delay 10-15, 10-17, 10-20, 10-21, 10-29, 10-33, 10-36, 10-38

u

UC Instruction 5-19, B-11
Unit Systems 2-3
User Defined Character Instruction, UC 5-19, B-11
User Units 2-3, 2-6, 8-2

Subject Index (Continued)

Using the Plotter with a Computer Mainframe, RS-232-C 10-2
Using the Plotter with a Personal Computer, RS-232-C 10-2
Using the Plotter with a Terminal 10-4, 10-9

V

VS Instruction 3-3, B-11
Velocity Select Instruction, VS 3-3, B-11

W

Window 2-1, 8-13
Window, Setting the 2-9, 8-13
Outputting the 2-10

X

XT Instruction 4-2, 8-3, B-11
Xoff Threshold Level 10-16, 10-30, 10-31
Xoff Trigger Character 10-16, 10-19, 10-20, 10-28, 10-31, 10-34
Xon Trigger Character 10-16, 10-19, 10-20, 10-30
Xon-Xoff Handshake ..... 10-14, 10-19, 10-28 thru 10-31, 10-36

Y

YT Instruction 4-2, 8-4, B-11

1 Getting Started

9 HP-IB Interfacing

2 Establishing Boundaries and Units

10 RS-232-C/CCITT V.24 Interfacing

3 Controlling the Pen and Plotting

11 HP-IL Interfacing

4 Enhancing the Plot

A An HP-IB Overview

5 Labeling

B Instruction Syntax

6 Digitizing

C Reference Material

7 Obtaining Information from the Plotter

8 Putting the Commands to Work

PART NO. 07470-90001
MICROFICHE NO. 07470-90051

HEWLETT PACKARD

OCTOBER 1984
PRINTED IN U.S.A.