PAC Hardware GmbH

Advanced PAC Screen V. 2.0
Programmers Manual

Beta Draft

Advanced PAC Screen V.2.0
Programmers Manual

Text Interface
Plotter Emulating Graphics Device
Centronics Interface

Written by Thomas Severin, Alexander Thul, Uwe Wildner, Eric Kamsties, Siegfried Hartmann

Limitation on Warranties and Liability

Even though PAC has testet the software described in this manual and reviewed its contents, neither PAC nor its software suppliers make any warranty or representation, either express or implied, with respect to this manual, their quality, performance, merchantability or fitness for any particular purpose.
As a result, this software and manual are sold "as is", and you the purchaser are assuming the entire risk to their quality and performance. In no event PAC or its software suppliers be liable for direct, indirect, incidental or consequential damages resulting from any defect in the software or manual, even if they have been advised of the possibility of such damages.

Copyright

Scaling Your Plot

General Plot Commands

Enhancing the Plot

Labelling

Obtaining Information from PACPLOT

Putting The Commands to Work - How to write Your own Programs

Extensions to the HPGL with Advanced Pac Screen 2.0

General Operation of the Advanced Pac Screen 2.0

The Advanced Pac Screen 2.0 is a multi purpose interface for the HP Interface Loop, HPIL. There are three functions included within the Advanced Pac Screen 2.0.

1. **Text Interface**
2. **Plotter Emulation**
3. **Centronics Interface**

The 3 different devices within the Advanced Pac Screen 2.0 are described in detail later. If there is just Advanced Pac Screen 2.0 connected to your computer it will recognize Advanced Pac Screen 2.0 as three different devices with 3 different Device ID´s , Accessory ID´s and so on. Each of the three devices can work independantly from each other. If there is the need you may adress all three devices at the same time and send data to the Text Interface, Plotter Interface and to the Printer simultaneously.

Getting started

After unpacking Advanced Pac Screen 2.0 you will see several sockets and a **POWER ON/OFF** switch in the front and rear panels.
On the backside is printed the serial number of your interface. Whenever you have problems you should give the serial number to your dealer. Besides there is printed some information about the pinning and function of the connectors. First you should plug in your video cable into the socket named **VIDEO OUT.** Thereafter plug in the plug of the power supply which came with Advanced Pac Screen 2.0 in the socket named **POWER SUPPLY**. Now switch on the Advanced Pac Screen 2.0. On the Video Screen appears the startup picture. Besides there is a LED on top of the Advanced Pac Screen 2.0 which should be on at the start. This LED indicates that Advanced Pac Screen 2.0 is idle at the moment. Whenever there is Loop operation this LED will flicker indicating that Advanced Pac Screen 2.0 is busy at the moment. If the LED is on but you don´t see any picture on the screen you should check the video monitor and the video cable.
If you bought the mouse together with the Advanced Pac Screen 2.0 you might plug it in the **MOUSEPORT** now. The presence of the mouse will be recognized by Advanced Pac Screen 2.0 automatically and the first line of the screen will change immediately.
After plugging in the cables for HPIL the Advanced Pac Screen 2.0 is ready for operation. How to send data to the Advanced Pac Screen 2.0 is described later in the chapters about the different devices.

Screen Partitioning and mouse operation

There are two devices within Advanced Pac Screen 2.0 which send their output to the screen (the Text interface and the Plotter interface). As there is just one screen available they have to share the screen. The screen is organized as two sheets of paper which lay one over the other. All the operations described below can be executed under program control using the command sequences of the TEXT INTERFACE described later.

Text Page and Graphic Page

One sheet contains the text sent to the text interface, the other sheet contains the plot you produce with the Plotter interface. After switching on the Advanced Pac Screen 2.0 the text sheet is visible. If you send data to the text interface the power on picture will disappear and your text is displayed.

Changing Text and Graphic Partitioning with the mouse

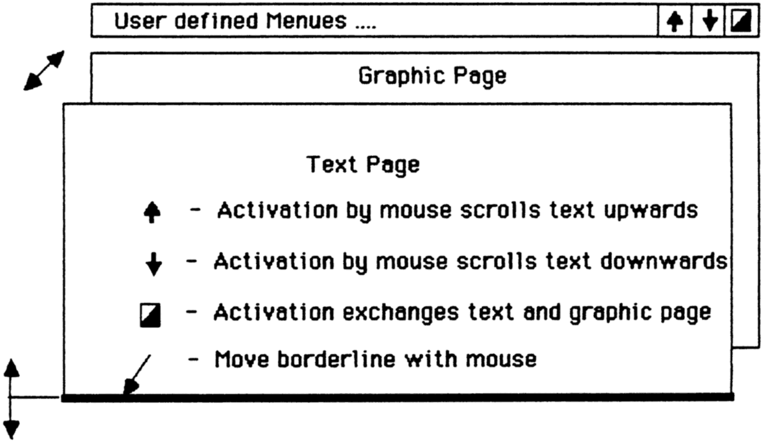
Advanced Pac Screen 2.0 offers the possibility to view both sheets at the same time. To that purpose you can move up the sheet laying on top of the screen. You move the mouse until the arrow shown on the screen is at the bottom, press the mouse button and shove the mouse upwards. A horizontal line appears on the screen marking the border between text and graphic sheet. When you release the mouse button the border will stay at that position until you repeat this procedure.

Exchanging Text and Graphic Page with the mouse

You may alter the page which lays on top of the screen using the mouse. To that purpose you move the mouse arrow to the top right corner of the screen. There are three signs displayed. Two arrows up and down and a **shaded rectangle**. If you move the mouse arrow to that rectangle, press the button and release it again the two page will interchange their positions immediately. With that procedure you may swap the text and graphic pages very easily.

Scrolling the Text Memory with the mouse

If you edit large programs the 25 lines of the video screen are not sufficient. Therefore the Text memory within Advanced Pac Screen 2.0 is much bigger than 25 lines. Actually Advanced Pac Screen 2.0 has 125 lines of Text Memory from which are maximal 25 lines displayed. If you want to see what lays before the visible part position the mouse arrow on the **Arrow Down Sign** in the top right corner of the screen. Pressing the mouse button here will scroll the text down. Moving the mouse arrow on the **Arrow Down Sign** in the top right corner of the screen and pressing the mouse button here will scroll the text up.



↑ - Activation by mouse scrolls text upwards

↓ - Activation by mouse scrolls text downwards

◪ - Activation exchanges text and graphic page

↙ - Move borderline with mouse

The TEXT INTERFACE of the Advanced Pac Screen 2.0

Within your Advanced Pac Screen 2.0 there is a powerful TEXT INTERFACE with 125 lines of text memory included. From the IL Loop the Text Interface is looked at as an extra device with following specifications:

DEVICE ID	-	PACTEXT
ACCESSORY ID	-	48

According to your calculator there are several ways to address the Text interface.

Using the **HP 71** as controller:
The first thing to do after switching on the HP 71 is executing a **RESET HPIL @ RESTORE IO** Instruction. Thereafter exist 3 ways to address the Text interface.

```
DISPLAY IS :PACTEXT
DISPLAY IS :%46
DISPLAY IS :3  ( only if APS is the first unit in the loop)
```

All strings sent to the loop using the DISP statement will be sent to the Text interface.

Using the **HP 41** as controller:

1	**ENTER**	the ´1´ is appropriate only if the
XEQ	**SELECT**	Advanced Pac Screen 2.0 is the 1.unit
		in the IL Loop.

All strings sent to the loop using the OUTA statement will be sent on to the Text interface.

Using the **HP 75** as controller:
The first thing to do after switching on the HP 75 is executing a **RESTORE IO** command.

Thereafter you should assign names to the devices connected to the IL Loop by executing **ASSIGN IO** . Now the HP 75 will inform you how many Devices are connected to the Loop and asks you about the names you want to assign to them. If the Advanced Pac Screen 2.0 is the first unit in the Loop you might name the first device "**TV**" for screen. Now you can adress the Text interface by **DISPLAY IS ":TV"**. All strings sent to the loop using the DISP statement will be sent to the Text interface.

Response to IL commands

Device ID	-	PACTEXT
Accessory ID	-	48
Status ID	-	0 normal condition
		100 mouse button pressed
		110 mouse button released after it has been pressed
		120 an active menu has been activated Information about line and column is available in the Output buffer of PACTEXT
Clear	-	Text screen and Text memory are cleared Menus are cleared Cursor is at Home position

Functions of the Advanced Pac Screen text device

Controlling the Cursor

ESC A	Moves cursor up one line
ESC B	Moves cursor down one line
ESC C	Moves cursor right one column
ESC D	Moves cursor left one column
ESC H	Cursor Home at position 0,0
ESC % xy	Cursor at position x,y x is the column ASCII-Code 0 bis 79 y is the row ASCII-Code 0 bis 24

Clear the Text Screen

ESC E	clear text screen and cursor home
ESC J	clear text screen from cursor position to end of screen

Cursor modes

ESC Q	displays the insertion cursor (underline)
ESC R	displays the replace cursor (block)

ESC <	Cursor off, no cursor is displayed
ESC >	Cursor on, cursor is displayed

Scroll commands

ESC S	Scroll screen up one line
ESC T	Scroll screen down one line
ESC a	Softscroll (Default)
ESC b	Hardscroll, is very fast

Read from Text Screen

ESC ? xyz	Read string with length z from cursor position x,y x is the column CHR$(0 .. 79) y is the row CHR$(0 .. 24) z is the length of the string CHR$(0 .. 255)

Soft Reset of the Advanced Pac Screen 2.0

ESC CHR$(127) and **ESC CHR$(255)**	Advanced Pac Screen 2.0 is set back in the power on state. This command has exactly the same effect as switching OFF and ON the Advanced Pac Screen 2.0 .

Screen partitioning

Advanced Pac Screen 2.0 offers the posibility to display text and graphic page at the same time. The physical screen is parted by a horizontal borderline which you can move with the mouse. You also can change the page which lays on top with the mouse. How to do that refer to **Screen partitioning and mouse** operation at the beginning of this manual.
When there is a mouse connected to Advanced Pac Screen 2.0 **the first two lines of the screen are needed for mouse operation.** There is no way to remove these two lines once there is a mouse connected to Advanced Pac Screen 2.0 .

ESC s Text and Graphic page are exchanged. This command has exactly the same effect as clicking on the shaded rectangle using the mouse.

ESC g The graphic page will be put on top of the screen.

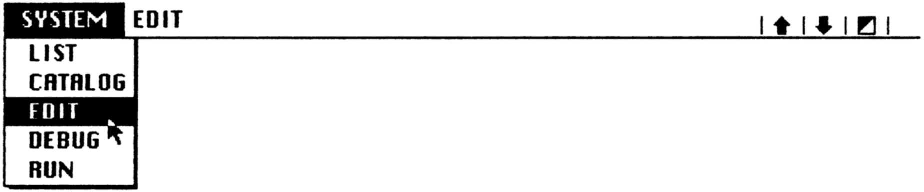
ESC t The text page will be put on top of the screen.

ESC w n The border between text and graphic page is set to Line n (2..24). You can move the border with the mouse as well.

Using Mouse and Menues with Advanced Pac Screen 2.0

The Mouse at your APS

The new **Advanced PAC Screen 2.0** provides the feature of connecting a mouse to it. That offers a very comfortable way of working with your computer. Moreover Advanced Pac Screen 2.0 supports the menu system. Menues are lists of words that are cataloged under headers. After defining them, these headers are visible all the time until you switch off Advanced Pac Screen 2.0 . But you can use the menues only if you have a mouse that fits into the Advanced Pac Screen 2.0 mouse connector. There are many applications for the menu system, like controlling programs and functions. There is a demo program supplied in this handbook which shows the use of the mouse. A menue has always the following structure:



When you have defined the menues (how to do that see below) , you can show one and select an entry using your mouse.

Use of the mouse

If there is a mouse connected to APS the menue headerline will appear in the first line on the screen after switching on APS . (If not yet please move the mouse a little bit.) Now APS has recognized the mouse and the screen has been splitted in two windows. Actually there are three ones but only the upper window is permanently visible. It is used for the menue headerline and sometimes for a graphic´s status line one line below. The other 23 lines of the screen are used for the graphic or the text window. It is about you to define what is visible. you can put graphic page below and overlay it with 10 lines of text or what you want. If you have no mouse the defining could only be done by sending escape-sequences to the APS ,if you have one you can also use your mouse to do this. Just press the button of your mouse and move it across the border between the windows (there is a line to indicate this border) and the border will follow the movements of your mouse (but don´t hurry or it probably won´t work). Upon this, there are these three little characters at the right end of the headerline. There is an arrow up , an arrow down and a slanting divided rectangle. These are system menues that allows you to cause actions that are usually caused by escape-sequences.

system menues	action
arrow up	scrolls up
arrow down	scrolls down
rectangle	swaps windows

If you place the arrow on one of these characters and press the button the specified action will be executed. This is a much more comfortable way of rushing through listings or alter working between graphics and text.

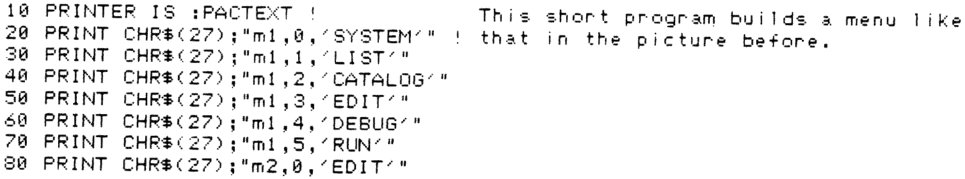
For programming the menues you need the following escape sequences:

<ESC> **m x,y 'string'** where x is an integer from 1 to 10 that specifies the number of the menue

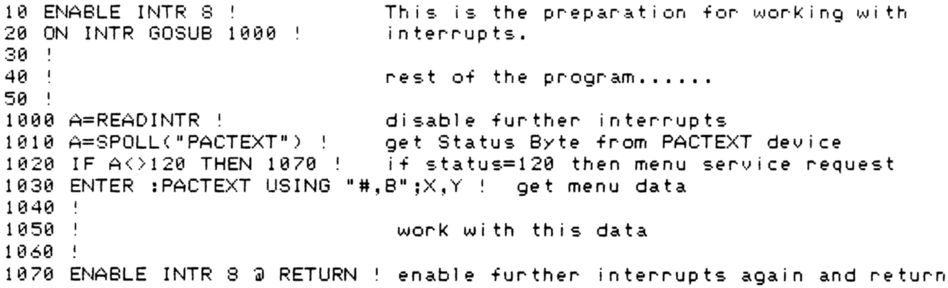
y is an integer from 0 to 20 that specifies the row of the selected menue

string is any literal string without single quotes up to a length of 20 characters (else it will be truncated to 20)

You need this escape sequence to define a specified entry of a menue. With a y-value of 0 you define a menue header that will appear in the menu headerline. That is the line that appears if a mouse is detected by the APS and that is visible all the time. All the other menue entries are submenue entries that will say you can see them only if you have selected the menue where it is in. That works like this : You move the arrow on the screen on one of the menue headers by moving the mouse. Now while you press the button of the mouse the header will be inverted and, if there is any, the submenue that belongs to this header will be shown. For selecting a menue entry you only have to drag the arrow on the entry and release the button. The selected entry will flash a few times, the coordinates (the x and the y value you used to define it) will be put on the loop and a loop interrupt will be created. This interrupt offers you a wide variety of facilities of programming. So you can use it to create your own operating system that is controlled with the mouse or a kind of full screen editor with mouse support. The main program part to work with this mechanism looks like this:



```
10 PRINTER IS :PACTEXT !                 This short program builds a menu like
20 PRINT CHR$(27);"m1,0,'SYSTEM'" !  that in the picture before.
30 PRINT CHR$(27);"m1,1,'LIST'"
40 PRINT CHR$(27);"m1,2,'CATALOG'"
50 PRINT CHR$(27);"m1,3,'EDIT'"
60 PRINT CHR$(27);"m1,4,'DEBUG'"
70 PRINT CHR$(27);"m1,5,'RUN'"
80 PRINT CHR$(27);"m2,0,'EDIT'"
```



```
10 ENABLE INTR 8 !              This is the preparation for working with
20 ON INTR GOSUB 1000 !         interrupts.
30 !
40 !                            rest of the program......
50 !
1000 A=READINTR !               disable further interrupts
1010 A=SPOLL("PACTEXT") !       get Status Byte from PACTEXT device
1020 IF A<>120 THEN 1070 !      if status=120 then menu service request
1030 ENTER :PACTEXT USING "#,B";X,Y !   get menu data
1040 !
1050 !                           work with this data
1060 !
1070 ENABLE INTR 8 @ RETURN ! enable further interrupts again and return
```

```
10 ! There are easier ways to get data from menu activation.
20 ! That is to check menu activation in a loop or just busy waiting.
30 A=SPOLL("PACTEXT") !      To handle the menues in a loop just test the
40 IF A=120 THEN GOSUB 100 ! Status Byte everytime and jump to a subroutine
50 !                         if there is some data.
60 !
70 !                         do some other stuff
80 !
90 GOTO 30 !                 end of the loop
100 ! subroutine to work with menu data
110 RETURN
120 ! ***********************************************************************
130 !                        The other way is busy waiting:
140 A=SPOLL("PACTEXT") !     check status byte if there is data available
150 IF A<>120 THEN 140 !     return to line 140 if there is no data
160 !
170 !                        do some stuff
```

To make the mouse an allround instrument you have a software switch to determine that the mouse creates a loop interrupt any time the button will be pressed and released.That is the escape sequence:

ESC x	sets the switch on and
ESC y	sets it off

This feature is useful to use the mouse i.e. for placing the cursor at the place determined by the mouse. Also the marking of parts of the text is a application that is possible to implement with this system.

The Plot Interface of the Advanced Pac Screen 2.0

From the IL Loop the Plot Interface is looked at as an extra device with following specifications:

DEVICE ID	-	PACPLOT
ACCESSORY ID	-	96

According to your calculator there are several possibilities to adress the Plot interface.

Using the **HP 71** as controller:
The first thing to do after switching on the HP 71 is executing a **RESET HPIL @ RESTORE IO** Instruction. Thereafter exist 3 possibilities to adress the Plot interface.

PRINTER IS :PACPLOT
PRINTER IS :%96
PRINTER IS :2 (only if APS is the first unit in the loop)

All strings sent to the loop using the PRINT statement will be sent on to the Plot Interface

Using the **HP 41** as controller:

2	**ENTER**	the '2' is appropriate only if the
XEQ	**SELECT**	Advanced Pac Screen 2.0 is the 1.unit in the IL Loop.

All strings sent to the loop using the OUTA statement will be sent on to Plot interface.

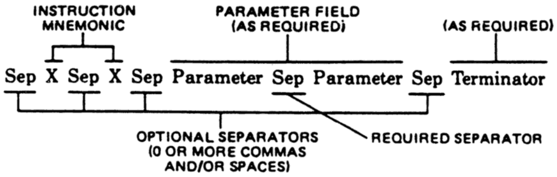
Using the **HP 75** as controller:
The first thing to do after switching on the HP 75 is executing a **RESTORE IO** command. Therafter you should assign names to the devices connected to the IL Loop by executing **ASSIGN IO** . Now the HP 75 will inform you how many Devices are connected to the Loop and asks you about the names you want to assign to them. If the Advanced Pac Screen 2.0 is the first unit in the Loop you might name the second device "**PL**" for plotter. Now you can address the Plot interface by **PRINTER IS ":PL"**. All strings sent to the loop using the PRINT statement will be sent on to the Plot interface.

Response to IL commands

Device ID	-	**PACPLOT**		
Accessory ID	-	**96**		
Status ID	-	**Bit 1**	set	Digitized points available
			reset	No data in Output Buffer
		Bit 4	set	Error occured previously
			reset	On executing the Output Error ,OE
Instruction				
		Bit 6	set	Data available after Hardcopy Commands HT, LO
			reset	On executing a serial poll operation
Clear	-	The Graphic screen is cleared		

The HP-Graphics Language

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 and/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. Some instructions will execute immediately after the mnemonic or at last required parameter is received. When this is the case, the designation for the terminator is shown in parantheses in the syntax description. The syntax is shown in the next paragraph.


INSTRUCTION MNEMONIC
PARAMETER FIELD (AS REQUIRED)
(AS REQUIRED)
Sep X Sep X Sep Parameter Sep Parameter Sep Terminator
OPTIONAL SEPARATORS (0 OR MORE COMMAS AND/OR SPACES)
REQUIRED SEPARATOR

The HP-GL syntax

Some instruction 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. You may send it with the CHR$(3) Basic command to the PACPLOT device. 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 parameter in integer format between -32768.0000 and +32767.9999. 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 -a combination of text, numeric expressions, or string variables . Refer to The Label Instruction, LB, 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 parantheses in the syntax descriptions.

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

mnemonic

For readability, the mnemonic is shown uppercase 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 Advanced PAC Screen's HP-GL instruction set.

Instruction		Descripticn
AA	X [i/sd], Y [i/sd], arc angle [i] (,chord angle [i])	Arc absolute
AR	X [i/sd], Y [i/sd], arc angle [i] (,chord angle [i])	Arc relative
CA	n [i]	Designate alternate set n
CI	radius [i/sd] (,chord angle [i])	Circle
CP	spaces [d], lines [d]	Character plot
CS	n [i]	Designate standard set n
DC		Digitize clear
DF		Set default values
DI	run [d], rise [d]	Absolute direction
DP		Digitize point
DR	run [d], rise [d]	Relative direction
DT	c [c]	Define label terminator
EA	X [i/sd], Y [i/sd]	Edge rectangle absolute
ER	X [i/sd], Y [i/sd]	Edge rectangle relative
EW	radius [i/sd], start angle [i], sweep angle [i] (,chord angle [i])	Edge wedge
FT	type [i] (,spacing [sd] (,angle [i]))	Fill type
IM	e [i] (,s [i] (,p [i]))	Input e, s, and p masks
IN		Initialize
IP	$P1_x$ [i], $P1_y$ [i] (,$P2_x$ [i], $P2_y$ [i])	Input P1 and P2
IW	X_{lo} [i], Y_{lo} [i], X_{hi} [i], Y_{hi} [i]	Input window
LB	c . . . c [c]	Label ASCII string
LT	t [d] (,1 [d])	Designate line type and length
OA	[i return]	Output actual position and pen status
OC	[i/sd return]	Output commanded position and pen status
OD	[i return]	Output digitized point and pen status
OE	[i return]	Output error
OF	[i return]	Output factors
OH	[i return]	Output hard-clip limits
OI	[c return]	Output identification
OO	[i return]	Output options
OP	[i return]	Output P1 and P2
OS	[i return]	Output status
OW	[i return]	Output window
PA	X [i/sd], Y [i/sd] (, . . .)	Plot absolute
PD	(X [i/sd], Y [i/sd] (, . . .))	Pen down
PR	X [i/sd], Y [i/sd] (, . . .)	Plot relative
PS	paper size [i]	Paper size
PT	thickness [d]	Pen thickness
PU	(X [i/sd], Y [i/sd] (, . . .))	Pen up
RA	X [i/sd], Y [i/sd]	Shade rectangle absolute
RO	n [i]	Rotate coordinate system
RR	X [i/sd], Y [i/sd]	Shade rectangle relative
SA		Select alternate character set
SC	X_{min} [i], X_{max} [i], Y_{min} [i], Y_{max} [i]	Scale
SI	width [d], height [d]	Absolute character size
SL	tanϕ [d]	Absolute character slant (from vertical)
SM	c [c]	Symbol mode
SP	n [i]	Select pen
SR	width [d], height [d]	Relative character size
SS		Select standard character set
TL	tp [d] (,tn [d])	Tick length
UC	(pen [i],) X [d], Y [d], pen [i] (, . . .)	User defined character
VS	v [d]	Select velocity v
XT		X-axis tick
YT		Y-axis tick

[c] = character format

[d] = decimal format, −128.0000 to +127.9999

[i] = integer format, −32 768 to +32 767

[sd] = scaled decimal format, −32 768.0000 to +32 767.9999

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 PACPLOT; 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. Don not send the quotation marks to PACPLOT.

Longer examples are given as programs or program segments in BASIC.
The programs will run only if PACPLOT has been defined as the system printer. Unless specific mention is made in the text, the BASIC used is that of the HP-71B . 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 PACPLOT especially if you are using the HP-75. Check with the nearest HP dealer or HP Sales and Support Office.

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.

The Default Instruction, DF

DESCRIPTION The default instruction, DF, set certain plotter functions to a predefined start.

USES The instruction can be used to return PACPLOT 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 follewing plotter functions to the conditions shown in the following table.P1 and P2 are not changed.

Default Conditions

Function	Equivalent Instructions	Conditions
Plotting mode	PA;	Absolute (PA)
Relative character direction	DR1,0;	Horizontal (DR1,0)
Line type	LT;	Solid line
Line pattern length	LT;	4% of the diagonal distance between P1 to P2
Input window	IW;	Set to current hard-clip limits

Relative character size	SR;	Width = 0.75% of ($P2_x - P1_x$)
		Height = 1.5% of ($P2_y - P1_y$)

Symbol mode	SM;	Off
Tick length	TL;	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	CS0;	Set 0
Alternate character set	CA0;	Set 0
Character set selected	SS;	Standard
Character slant	SL0;	0 degrees
Mask value	IM 223,0,0	223,0,0
Digitize clear	DC;	Off
Scale	SC;	Off
Pen velocity	VS;	38.1 cm/s (15 in./s)
Label terminator	DT ETX	ETX (ASCII decimal equivalent 3)
Chord angle	—	Set to 5 degrees
Fill type	FT;	Set to type 1, bidirectional solid fill
Fill spacing	FT;	1% of the diagonal distance between P1 and P2
Fill angle	FT;	Set to 0 degrees

The Initialize Instruction, IN

DESCRIPTION The initialize instruction, IN, returns PACPLOT's graphics conditions to the initial power-on state by program control. This instruction has no effect on the plot device's environment state.

USES The instruction can be used to return PACPLOT 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.

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 PACPLOT off and then on again. The initialize command sets PACPLOT 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 = 0,0 and P2 = 10224,7296 (paper size Din A4 with mouse).
- All HP-GL errors are cleared. Bit position 3 of the output status byte is set to true (1) indicating PACPLOT has been initialized. (This bit is cleared by OS.)

The Input Mask Instruction, IM

DESCRIPTION The input mask instruction, IM, controls the conditions under which HP-GL error status is reportet.

USES This instruction can be used to change the conditions under which HP-GL error status is reported.

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

EXPLANATION The S- and P-masks are of no use and are ignored if present. Only the E-mask is used .

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 . 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 + 4 + 2 + 1) will specify that all errors except error 6 will set the error bit in the status byte whenever they occur. Error 6 will not set the error bit 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.

PACPLOT, 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.

The Paper Size Instruction, PS

DESCRIPTION The paper size instruction, PS, provides the means to programmatically A3 and A4 paper size.

USES This instruction can be used to change the paper sizes programmatically.

SYNTAX **PS** paper size terminator

EXPLANATION A parameter in the range of 0-3 selects A3 size paper, and a parameter in the range of 4-127 selects A4 size paper.

If the PS instruction sets the paper size to the current size, the instruction is ignored. Specifying out-of-range parameters sets error3 and the instruction is ignored.

This instruction is included for compatibility with the 7275 only. Because the video screen allways has the same size, the PS instruction changes the values of P_1 and P_2 , the hardclip limits, and the limits of the window .

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.

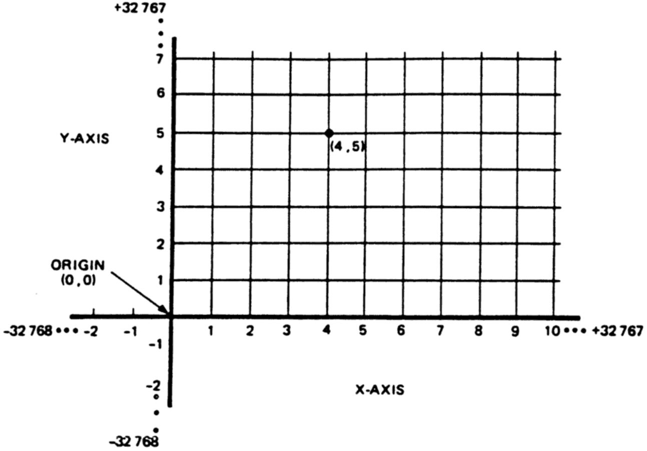
Scaling Points - two points which are usually placed at bottom left and top right of the screen. These points are assigned the user-unit values specified by the parameters of the scaling instruction SC.

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 PACPLOT. 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 of a plotter is that part of the paper in which the pen can draw. With the HP 7475 plotter it is possible to select different paper sizes.
As a video screen always has the same size a paper size instruction only changes the physical dimension of a Plotter Unit and the Hard Clip Limits.



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 approximatly 1000 plotter units per inch. One plotter unit is the smallest move the plotter can make. The resolution of a Video screen is much lower. The smallest space distinguishable on the screen equals 16 plotter units. When the paper switch is set to A4 and a mouse is connected, the plotting area contains 10 224 plotter units in X and 7296 plotter units 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 PACPLOT. When plotting in plotter units, only integer values are used; parameters are truncated to integers. Refer to The Plot Absolute Instruction, PA.

At power on and whenever an IN command is sent to PACPLOT, the scaling point P1 is set to 0,0 plotter units and the scaling point P2 is set to 10224,7296 plotter units.

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. You will probably use the SC instructions and user units for most plots.

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 or directions of labels when relative character size or direction is in effect; and to set P1 and P2 back to their default locations

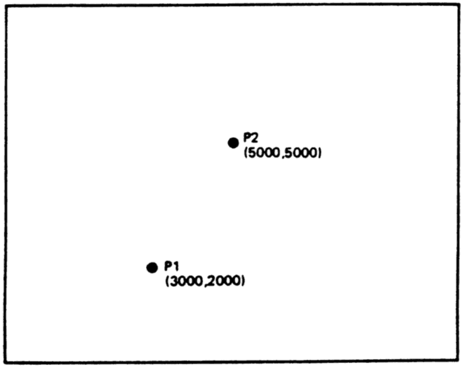
SYNTAX IP P1x,P1y (,P2x,P2y) (terminator)
or
IP (terminator)

EXPLANATION The new coordinates of P1 ands 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 $0 \le X \le 10\,224$ if the paper switch is set to A4; and $0 \le Y \le 7296$. 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 32767 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 0,0,10224,7296 regardless of the paper switch setting. If no mouse is connected 7968 is used for $P2_y$.
Upon receipt of a valid IP command, bit position1 of the output status word is set true (1).
Upon power on 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.

```
"IP3000,2000,5000,5000;"
```



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.

SYNTAX OP (terminator)

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

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

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

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 yout 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-coordinater 0 to 10, you might scale the X-axis 0to 14 so the first and last bars are nor 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 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.

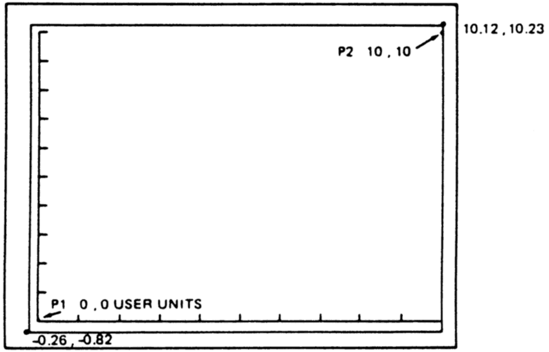
Specifying X_{max}= X_{min} or Y_{max}= Y_{min} or parameters less than -32768 or greater than 32 767 will turn scaling off. An SC command must have four or no parameters. Otherwise, error 2 will be generated. An SC command which generates an error is ignored and the scaling does not change.

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 impossible to scale an area such that P1 or P2 are assigned values larger than 32 767 or less than-32 765. One way to plot data with values beyond this limits is to reduce your data to acceptagle ranges by an arithmatic process before sending it to PACPLOT. 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 Din A4.

"IP; 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 dataarea. It is also useful when hatching (shading) ractangular areas.

SYNTAX IW $X_{lower\ left}$, $Y_{lower\ left}$, $X_{upper\ right}$, $Y_{upper\ right}$ (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.

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 10224 or 14459 for X (depending on the the setting of the paper switch) and less than 7968 or 7296 for Y (If there is a mouse connected) . 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 10 318 or 14 459 for X and 7296 or 7968 for Y. If $X_{lower\ left}$ is greater than $X_{upper\ right}$ or $Y_{lower\ left}$ is greater than $Y_{upper\ right}$, 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 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 any plotting will occur.

USES The instruction can be used to determine the area in which any plotting will occur.

SYNTAX OW (terminator)

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

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

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

where [TERM] is the output terminator for your system. See terms You should understand in chapter...

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

With mouse

A3	A4
$0 \leq X \leq 14\,459$	$0 \leq X \leq 10\,224$
$0 \leq Y \leq 10\,318$	$0 \leq Y \leq 7\,296$

Without mouse

A3	A4
$0 \leq X \leq 14\,459$	$0 \leq X \leq 10\,224$
$0 \leq Y \leq 11\,268$	$0 \leq Y \leq 7\,968$

If $Xl_{\text{ower left}}$ 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.

Advanced Programming Tips

Many software packages read P1 ans P2 and use these points to define the maximum plotting area. You may want to obtain the largest plot possible on PACPLOT . 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 will set P1 and P2 to these points, so that the largest area possible is used for plotting.

THE OUTPUT HARD-CLIP LIMITS INSTRUCTIONS, OH

DESRIPTION The output hard-clip limits instruction, OH, is used to output the lower-left (LL) and upper-right (UR) coordinates of the current hard-clip limits.

USES This instruction can be used with the IP instruction to determine and make use of the maximum available plotting area.

SYNTAX OH terminator

EXPLANATION After an OH instruction is received, the plotter will output the LL and UR coordinates in plotter units as four ASCII integers in the following form:

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

where TERM is the otput terminator for your system.

The APS suppresses leading zeros and positive signs. The Input P1 and P2 Instruction, IP, can be used to relocate P1 and P2 to the maximum plotting area as determined by the OH instruction.

THE ROTATE COORDINATE SYSTEM INSTRUCTION, RO

INSTRUCTION The rotate coordinate system instruction, RO, programmatically rotates the APS unit/user-unit coordinate system 90 degrees.

USES The instruction is implemented for compatibility only.

SYNTAX RO (angle in degrees) terminator
or
RO terminator

EXPLANATION The only allowable parameters are 0 and 90. The instruction RO90; rotates the current coordinate system 90 egrees from its default orientation as shown in the following diagrams for A4 and A3 paper size. Rotations are not cumulative, and the rotate function can only be toggled on and off. The instruction RO0; is the same as RO; and turns off the rotate function.

When RO90; instruction is executed, P1 and P2 retain their current coordinate values and may therefore be rotated outside the hard-clip limits. The current input window is also rotated, and any portion that is rotated outside of the hard-clip limits is clipped to the hard-clip limits. The size of the clipped input window can be determined by executing the OW instruction. The input window can be expanded to the hard-clip limits and P1 and P2 can be defaulted to their rotated default coordinate values using the instructions IW and IP without parameters.
The 0,0 origin point moves when the coordinate system is rotated, but the physical size and location of the hard-clip limits are not affected. However, the defined lower-left (LL) and upper-right (UR) corners of the hard-clip limits are rotated to maintain the same relationship with respect to the 0,0 origin point. The coordinate values for UR are determined by paper size and the state of the rotate function; but the coordinate values for LL will always be 0,0 regardless of paper size and the state of the rotate function. The current plotter unit coordinate values for LL and UR can be obtained by executing the OH instruction.

When the coordinate system is rotated, the logical pen positon is changed to correspond with the current physical pen position. The coordinate values of the new logical pen position can be obtained by executing either an OA or OC instruction after the rotate instruction is executed.

Specifying parameters other than 0 or 90 sets error 3 and the instruction is ignored. If you specify too many parameters, the instruction is executed with the first parameter, error 2 is set, and the rest of the parameters are ignored.

The initialize instruction, IN, defaults the rotation state to 0 degrees.

HP-GL Instructions Covered :

SP	The Select Pen Instruction
VS	The Velocity Select Instruction
PU,PD	The Pen Up,Down Instruction
PA	The Plot Absolute Instruction
PR	The Plot Relative Instruction
CI	The Circle Instruction
AA	The Arc Absolute Instruction
AR	The Arc Relative Instruction
FT	The Fill Type Instruction
PT	The Pen Thickness Instruction
RA	The Shade Rectangle Absolute Instruction
EA	The Edge Rectangle Absolute Instruction
RR	The Shade Rectangle Relative Instruction
ER	The Edge Rectangle Relative Instruction
EW	The Edge Wedge 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 espressed in plotter units.

The Velocity Select Instruction, VS

By plotting with a plotter this Instruction is used to obtain better results by slowing down plotting speed. Within a Video Interface there is no use for this command. A VS command is ignored by PACPLOT. No error is set unless the parameters are out of Range.

The Pen Thickness Instruction, PT

Not implemented, refer to the Velocity Select Instruction, VS.

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 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.

The Select Pen Instruction, SP

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

USES The instruction is used change drawing mode during p rogram execution

SYNTAX

```
SP  pen number, drawing mode (terminator)
or
SP  (terminator)
```

EXPLANATION The pen parameter must be in the range of 0 to 6. It will be ignored by PACPLOT. The parameter drawing mode is added to the HPGL to select different plot modes which are helpful when plotting on a video screen. It must be in the range of 0 to 3. This parameter is not valid with the HP 7475 plotter. It will cause an error when send to the plotter.

0 - Default pen ADD modus, lines occur white on a black screen. Plotting dotted lines intersecting lines won´t be cleared at the point of crossing.

1 - Replace mode , same as Add mode, but intersecting lines will be overwritten.

2 - Inverse pen, all locations on the screen moved over change their color, i.e. black becomes white and vice versa.
This is very helpfull to write a notice over a plot which shall be removed afterwards. By writing it twice with the inverse pen you don´t destroy your plot.

3 - Erase pen, white places on the screen moved over with the pen will become black

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_1coordinate(,X_2coordinate,
Y_2coordinate,...,$X_n$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.

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 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 plottet.

The X-coordinate specifies, in either plotter or user units, the absolute X-location to which the pen will move. The Y-coordinate specifies, in either plotter 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.

for example PD X_1,Y_1,X_2,Y_2PUX_3,Y_3PDX_4,Y_4;

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 Winow Instruction, IW.

PACPLOT 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 occurences 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.
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 this point, the pen assumes its programmed (up or down) position. The pen then moves to the new point.

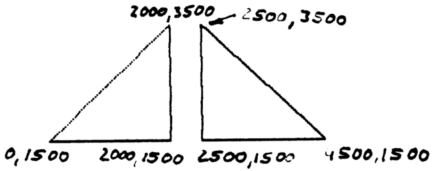
In type four, no pen movement occurs unless the straight line between the last and new point intersects the window. The X- and Y- coordinates of the current pen position are updated. If part of the vector is in the window area, the pen moves, pen up, to the point where the line between the last and the 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.

Since out-of-range points are discarded, PACPLOT 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

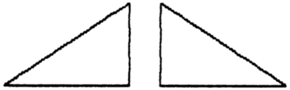
The following strings of HP-GL instructions, if sent to PACPLOT using a suitable output statement such as PRINT or OUTPUT, will draw two triangles .

```
10 PRINT "IN;SP1;"
20 PRINT "PA2000,1500,PD,0,1500,2000,3500,2000,1500,PU,2500,1500;"
30 PRINT "PAPD4500,1500,2500,3500,2500,1500,PU,10365,7721;"
```

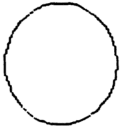


The next strings of HP-GL instructions scale the plotting area into user units 0 to 100 in each axis and again draws two triangles. Use an output statement implemented on your computer to send the strings to PACPLOT.

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



This final example scales a square plotting 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 PACPLOT 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 PACPLOT in scientific notation, which sets an error in PACPLOT.



```
5 RADIANS
10 PRINTER IS 'PACPLOT'
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 specifies 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 increments, 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 commend requires that both increments of an X,Y pair be given. An odd number of parameters will set an error conditon but all X,Y pairs which precede the unmatched parameters will be plotted.

The X-increment specifies, in either plotter units, the number of units the pen n 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 parameters determines the direction of movement; a positive value moves the pen in a positive direction and a negative value moves the pen in a 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 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 instrument 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 Instuction, IW. Drawing of vectors in relation to the window is as described under the PA instruction.

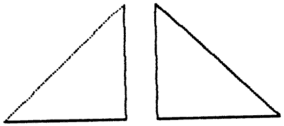
PACPLOT discards parameters which are out of range or whose plotter unit equivalent would be out of range if the indicated move were made.

Error 3 will be set (parameters 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 betwen -32 768 and 32 767.

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

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



PLOTTING WITH VARIABLES

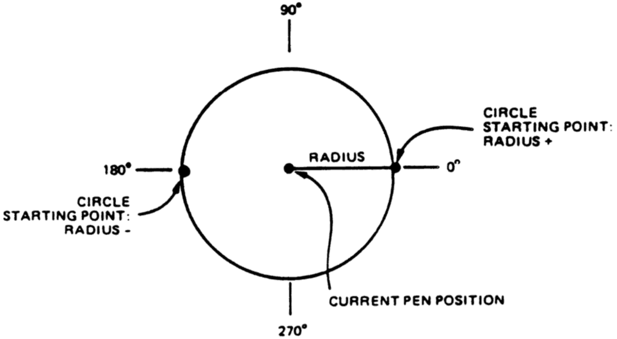
In many plotting applications, it is necessary to plot using variables rather than fixed numbers to define the X- and Y-coordinate values.The values of all HP-GL statement parameters have the same restrictions (integer or decimals in a valide range) when set as variables as when sent as literals (fixed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

THE CIRCLE INSTRUCTION, CI

DESCRIPTION The circle instruction,CI, provides the means to draw a circle of a specified radius and chord angle.

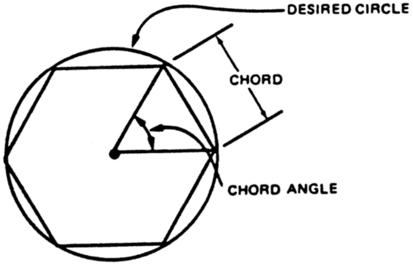
USES The instruction can be used to generate circles with a single command. All computations are internal to PACPLOT to reduce computer overhead.

SYNTAX CI radius (, chord angle) terminator.



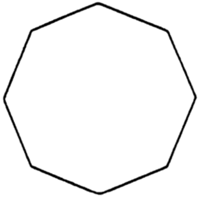
EXPLANATION The radius parameter can be a positive or negative number in integer format. Its sign defines the starting point of the circle: a circle with a positive radius starts at the 0-degree point; a circle with a negative radius starts at the 180-degree point. The current pen position is the center of the circle. If scaling is 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 PACPLOT so that al chords are the same length. The sign of the parameter is ignored, except to set the maximum in-range limit to -32 768 or +32767.

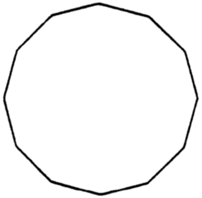


The most useful chord angle values range from 0 to 180; where 0 produces the smoothest circle and larger numbers progressively reduce the number of chords used. Values from 180 to 360 work just the opposite; i.e., larger numbers progressively increase the number of chords used and 360 produces the smoothest cicle. 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.

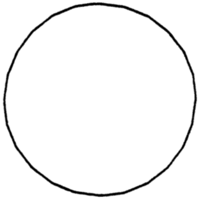
The following program shows the effect of different chord angles.



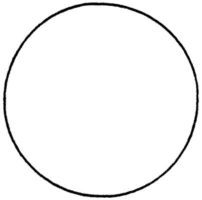
45 DEGREE CHORD ANGLE



30 DEGREE CHORD ANGLE



15 DEGREE CHORD ANGLE



5 DEGREE CHORD ANGLE

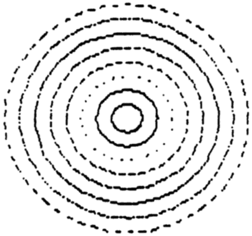
The circle instruction includes an automatic pen down feature. When a circle command is received, the pen lifts (if it was down), moves from the center of the circle to the circle starting point on the circumference, lowers the pen, draws the circle, then returns, pen up, to the center of the circle.After 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.

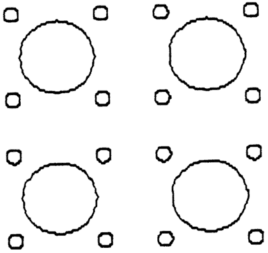
To demonstrate some of the features of the circle instruction, the following strings of HP-GL instructions draw various circles with different line types, radii, and starting points.

```
10 PRINT "IN;SP1;IP2650,1325,7650,6325;"
20 PRINT "SC-100,100,-100,100;"
30 PRINT "PA0,0;LT;CI10,5;LT0;CI-20,5;LT1;CI30,5;"
40 PRINT "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.

```
1 PRINTER IS 'PACPLOT'
10 PRINT "IN;SP1;IP2650,1325,7650,6325;"
20 PRINT "SC-1000,1000,-1000,1000;"
30 PRINT "PA-800,800;"
40 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 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.

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 positon, and its center 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.

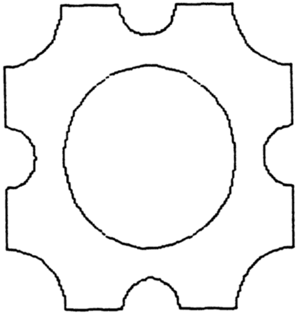
The chord angle parameter is in integer format and governs the smoothness of the arc in the same way as defined under the circle instruction, CI. The sign of the parameter is ignored, except to set the maximum in-range limit to -32 768 or +32 767. The default chord angle is 5 degrees.

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 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 'PACPLOT'
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 PACPLOT 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.

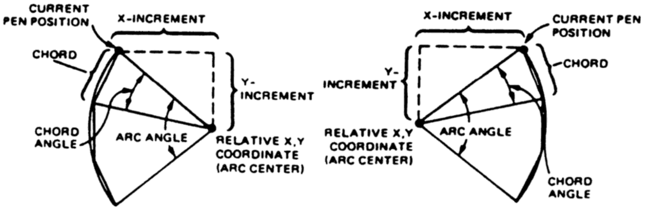
Lines 60 through 90 continue drawing the figure, clockwise, back to the point 0,20, and finish with the circle centered at the point 50,50.

THE ARC RELATIVE INSRTUCTION, 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.

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 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.

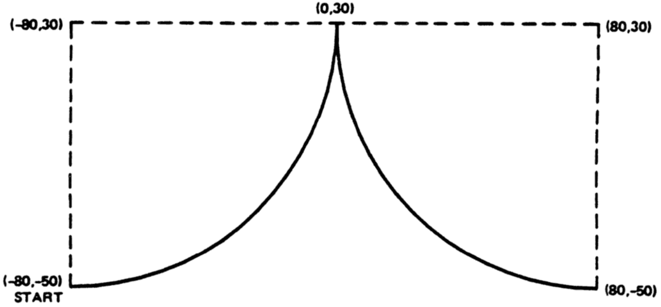
The chord angle parameter is in integer format and governs the smoothness of the arc in the same way as defined under the circle instruction, CI. The sign of the parameter is ignored, except to set the maximum in-range to -32 768 or +32 767. The default chord angle is 5 degrees.

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 commended, 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 integer in the range -32 768 to 32 767. Specifying out-of-range parameters sets error 3 and command is ignored.

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



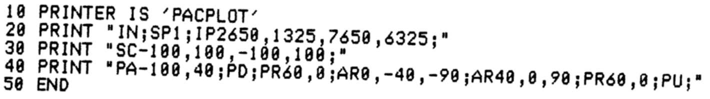
```
10 PRINTER IS 10
20 PRINT "IN;SP1;IP2650,1325,7650,6325;"
30 PRINT "SC-100,100,-100,100;"
40 PRINT "PA-100,40;PD;PR60,0;AR0,-40,-90;
          AR40,0,90;PR60,0;PU;"
```

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 'PACPLOT'
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-degee 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.

THE FILL TYPE INSTRUCTION,FT

DESCRIPTION The fill type instruction, FT, selects the type of area fill for use with an RA,or RR instruction.

USES The instruction can be used to enhance pie charts, bar charts, and other graphs with solid fill, parallel lines, or cross-hatching.

```
SYNTAX  FT   (type(,spacing(,angle)))terminator
        or
        FT  terminator
```

EXPLANATION There are five types of area fill:

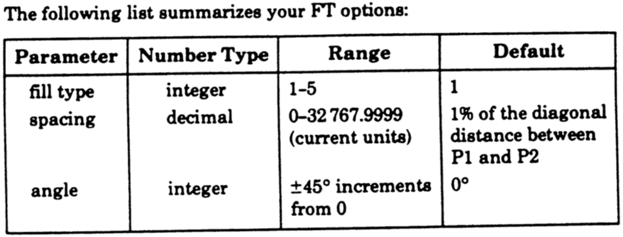
1. solid
2. solid
3. parallel-lines
4. cross-hatch
5. ignored

The fill type parameter should always be an integer number between one and four. If you do not specify a type, it will be defaulted to type one.

Spacing is the distance betwen parellel lines in the shade area. The units for spacing are interpreted as plotter units if scaling is off or as user units if scaling is on. The maximum allowable range is between 0 and 32 767. If you do not specify spacing, and this is the first FT instruction in your program, the spacing will be defaulted to 1% of the diagonal distance between P1 and P2.

If you do not specify spacing and this is not the first FT instruction in your program, the spacing specified in the previous FT instruction will be used. The spacing parameter is ignored for solid-fill types 1 and 2.
Determine the angle (line slant) using increments of 45 degrees starting from 0 degrees. Specifying a 0-degree angle will produce horizontal lines, a 90-degrees angle will produce vertical lines, and a 45-degree angle will produce angular lines. If you do not specify the angle and this is the first FT instuction in your program, the angle will be defaulted to 0 degrees. If you do not specify the angle and this is not the first FT instruction in your program, the angle specified in the previous FT instruction will be used.

The following list summarizes your FT options:



The following list summarizes your FT options:

Parameter	Number Type	Range	Default
fill type	integer	1-5	1
spacing	decimal	0-32 767.9999 (current units)	1% of the diagonal distance between P1 and P2
angle	integer	±45° increments from 0	0°

Specifing out-of-range parameters sets error 3 and the instruction is ignored. If you send too many parameters, error 2 is set, the instruction is executed with the first three parameters, and the rest of the parameters are ignored.

A default instruction, DF, or an initialize instruction, IN, will reset the fill type, spacing, and angle to default values.

THE SHADE RECTANGLE ABSOLUTE INSTUCTION,RA

DESCRIPTION The shade rectangle absolute instruction, RA, is used to define and shade a rectangle using absolute coordinates.

USES This instruction is used with the FT and PT instructions to fill a rectangle defined by the absolute points specified in the X- and Y-coordinate parameters. For an in-depth discussion of absolute plotting, see the explanation of The Plot Absolute Instruction, PA, in this chapter.

SYNTAX RA X-coordinate, Y-coordinate terminator

EXPLANATION The RA instruction requires that both X- and Y-coordinates be specified (coordinate pair). They are interpreted as plotter units if scaling is off or as user units if scaling is on. The current pen position is the starting point of the rectangle and the X- and Y-coordinates define the opposite corner of the rectangle. The maximum parameters are decimal numbers between -32 768.0000 and 32 767.9999. When scaling is off, the parameters are truncated to integers as follows:

- For positive numbers, the fractional portion is trunctated and the portion remains unchanged. For example, both 1234.4 and 1234.9 becomes 1234.

- For negative nmbers, the fractional portion is rounded up to the nextnegative integer. For example, both -1234.4 and -1234.9 become -1235.

An RA instruction with no parameters is ignored but no error is set. Specifying out-of-range parameters sets error 3 and the instruction is ignored. If you specify only one parameter, the instruction is ignored and error 2 is set. If you send too many parameters, the instruction is executed with the first two parameters, error 2 is set, and the rest of the parameters are ignored.

THE EDGE RECTANGLE ABSOLUTE INSTRUCTION, EA

DESCRIPTION The edge rectangle absolute instruction, EA, edges a rectangle defined in absolute coordinates.

USES This instruction draws the outline of a rectangle. It can be used with the RA instruction to outline a filled rectangle. For an indepth discussion of absolute plotting, see the explanation of The Plot Absolute Instruction, PA, located in this chapter.

SYNTAX EA X-coordinate, Y-coordinate terminator

EXPLANATION The EA instruction requires that both X- and Y-coordinates be specified (cordinate pair). They are interpreted as plotter units if scaling is off or as user units if scaling is on. The current pen position is the starting point of the rectangle and the X- and Y-coordinates define the opposite corner (diagonal endpoint) of the rectangle. The maximum parameters are decimal numbers between .32 768.0000 and 32 767.9999. When scaling is off, the parameters are truncated to integers as follows:

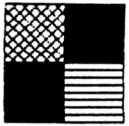
- For positive numbers, the fractional portion is trunctated and the next portion remains unchanged. For example, both 1234.4 and 1234.9 become 1234.

- For negative numbers, the fractional portion is rounded up to the next more negative integer. For example, both -1234.4 and -1234.9 become -1234.

An EA instruction with no parameters is not executed but no error is set. Specifying out-of-range parameters sets error 3 and the instruction is ignored. If you send only one parameter, error 2 is set and the instruction is ignored. If too many parameters are specified, then the instruction is executed with the first two parameters, error 2 is set, and the rest of the parameters are ignored.

The plotter will edge the designated rectangle, return the pen to the starting point, and restore the pen status upon completion of the instruction. The following BASIC program demonstrates the use of the EA, RA, and FT instructions.

```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SP1;PA5000,4000;"
25 PRINT "PD;"
30 PRINT "FT1;RA4000,3000;"
40 PRINT "EA4000,3000;"
50 PRINT "FT3,128,0;RA6000,3000;"
60 PRINT "EA6000,3000;"
70 PRINT "FT2;RA6000,5000;"
80 PRINT "EA6000,5000;"
90 PRINT "FT4,256,45;RA4000,5000;"
100 PRINT "EA4000,5000;"
110 END
```



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

20 initializes the plotter, selects a pen (pen 1), and sets the starting position.

30 selects pen thickness, fill type (solid fill, bidirectional), and sets the X,Y coordinates for the first rectangle.

40 selects a new pen (pen 3) and edges the first rectangle.

50 selects a new pen, a new fill type, and sets the X,Y coordinates for rectangle 2.

60 selects a new pen and edges rectangle 2.

70 selects a new pen, new fill type, and sets the X,Y coordinates for rectangle 3.

80 selects a new pen and edges rectangle 3.

90 selects a new pen, new fill type, spacing and angle, and sets the X,Y coordinates for rectangle 4.

100 selects a new pen and edges rectangle 4.

110 puts the pen back in the carousel.

THE SHADE RECTANGLE RELATIVE INSTRUCTION, RR

DESCRIPTION The shade rectangle relative instruction, RR, can be used to define and shade a rectangle using relative coordinates.

USES This instruction is used with the FT instruction to fill a rectangle defined from a point located relative to the present pen position. For an in-depth discussion of relative plotting, see the explanations of The Plot Relative Instruction, PR, located in this chapter.

SYNTAX RR X-increment, Y-increment terminator

EXPLANATION The RR instruction requires that both X- and Y-increment parameters be specified (coordinate pair). They are interpreted as plotter units if scaling is off or as user units if scaling is on. The current pen position is the starting point of the rectangle and the X- and Y-coordinates define the opposite corner (diagonal endpoint) of the rectangle. As with The Shade Rectangle Instruction, RA, the maximum parameters are decimal between -32 768.0000 and 32 767.9999. When scaling is off, the parameters are truncated to integers as follows:

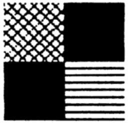
- For positive numbers, the fractional portion is trunctated and the integer portion remains unchanged. For example, both 1234.4 and 1234.9 become 1234.

- For negative numbers, the fractional portion is rounded up to the next negative integer. For example, both -1234.4 and -1234.9 become -1235.

An RR instruction with no parameters is ignored but no error is set. Specifying out-of-range parameters sets error 3 and the instruction is ignored. If you specify only one parameter, the instruction is ignored and error 2 is set. If too many parameters are sent, then the instruction is executed with the first two parameters, error 2 is set, and the rest of the parameters are ignored.

The rectangle is filled using the current pen and line type. At the completion of the instruction, the pen is returned to the original position. The following BASIC program, similar to the one used under the RA instruction, demonstrates the use of the RR and FT instructions.

```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SP1;PA3000,3000;"
30 PRINT "FT1;RR1000,1000;"
40 PRINT "PR1000,0;"
50 PRINT "FT3,128;RR1000,1000;"
60 PRINT "PR0,1000;"
70 PRINT "FT2;RR1000,1000;"
80 PRINT "FT4,256,45;RR-1000,1000;"
90 END
```



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

20 initializes the plotter, selects a pen (pen 1), and sets the starting position.

30 selects pen thickness, fill type (solid fill,), and sets the X,Y coordinates for the first rectangle.

40 moves the pen relative to its current location by the number of units specified by the X- and Y-parameters.

50 selects the fill type and spacing, and sets the X,Y coordinates for rectangle 2.

60 moves the pen relative to its current location by the number of units specified by the X- and Y- parameters.

70 selects the fill type and sets the X,Y coordinates for rectangle 3. Notice that you do not need to repeat the pen thickness for fill type 2 since it will remain in effect until you select a new pen or a new pen thickness.

80 selects the fill type, spacing, and angle and sets the X,Y coordinates for rectangle 4.

THE EDGE RECTANGLE RELATIVE INSTRUCTION, ER

DESCRIPTION The edge rectangle relative instruction, ER, edges a rectangle using relative plotting.

USES This instruction draws the outline of a rectangle. It can be used with the RR instruction to outline a filled rectangle. For an indepth discussion of relative plotting, see the explanation of The Plot Relative Instruction, PR, in this chapter.

SYNTAX ER X-coordinate, Y-coordinate terminator

EXPLANATION The ER instruction requires that both X- and Y-coordinates be specified (coordinate pair). They are interpreted as plotter units if scaling is off or as user units if scaling is on. The current pen position is the sarting point of the rectangle and the Y- and Y-coordinates define the opposite corner (diagonal endpoint) of the rectangle. As with The Edge Rectangle Absolute Instruction, EA, the maximum parameters are decimal numbers between -32 768.0000 and 32 767.9999. When scaling is off, the parameters are trunctated to integers as follows:

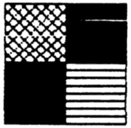
- For positive numbers, the fractional portion is trunctated and the integer portion remains unchanged. For example, both 1234.4 and 1234.9 become 1234.

- For negative numbers, the fractional portion is rounded up to the next negative integer. For example, both -1234.4 and -1234.9 become -1235.

An ER instruction with no parameters is not executed but no error is set. Specifying out-of-range parameters sets error 3 and the instruction is ignored. If you send only one parameter, error 2 is set, and the instruction is ignored. If too many parameters are specified, then the instruction is executed with the first two parameters, error 2 is set, and the rest of the parameters are ignored.

PACPLOT will edge the designated rectangle, return the pen to the starting point, and restore the pen status upon completion of the instruction. The following BASIC program demonstrates the use of the ER, RR, and FT instructions.

```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SP1;PA3000,3000;"
25 PRINT "PD;"
30 PRINT "FT1;RR1000,1000;"
40 PRINT "ER1000,1000;"
50 PRINT "PR1000,0;"
60 PRINT "FT3,128;RR1000,1000;"
70 PRINT "ER1000,1000;"
80 PRINT "PR0,1000;"
90 PRINT "FT2;RR1000,1000;"
100 PRINT "ER1000,1000;"
110 PRINT "FT4,256,45;RR-1000,1000;"
120 PRINT "ER-1000,1000;"
130 END
```



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

20 initializes the plotter, selects a pen (pen 1) and sets the starting position.

30 selects pen thickness, fill type (solid fill, bidirectional), and sets the X,Y coordinates for the first rectangle.

40 selects a new pen (pen 3) and edges the first rectangle.

50 moves the pen relative to its current location by the number of units specified by the X- and Y-parameters.

60 selects a new pen, a new fill type, and sets the X,Y coordinates for rectangle 2.

70 selects a new pen and edges rectangle 2.

80 moves the pen relative to its current location by the number of units specified by the X- and Y- parameters.

90 selects a new pen, a new fill type, and sets the X, Y coordinates for rectangle 3.

100 selects a new pen and edges rectangle 3.

110 selects a new pen, a new fill type, spacing and angle, and sets the X,Y coordinates for rectangle 4.

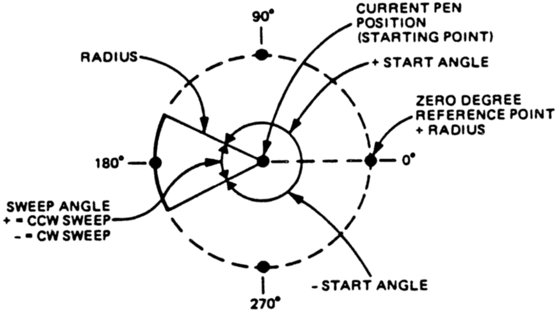
120 selects a new pen and edges rectangle 4.

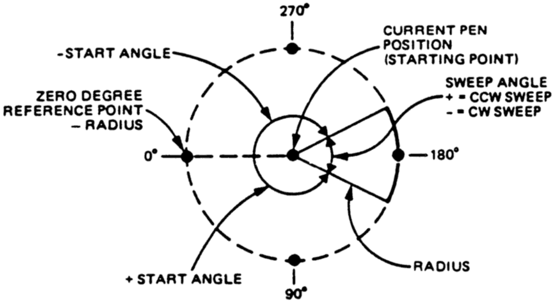
THE EDGE WEDGE INSTRUCTION,EW

DESCRIPTION The edge wedge instruction, EW, is used to edge any arc segment of a circle of a specified radius.

USES This instruction is used to produce individual arc segments that can be combined to create a pie chart.

SYNTAX EW radius, start angle, sweep angle(,chord angle) terminator


90°
CURRENT PEN POSITION (STARTING POINT)
RADIUS
+ START ANGLE
ZERO DEGREE REFERENCE POINT + RADIUS
180°
0°
SWEEP ANGLE + = CCW SWEEP - = CW SWEEP
- START ANGLE
270°


270°
CURRENT PEN POSITION (STARTING POINT)
-START ANGLE
ZERO DEGREE REFERENCE POINT – RADIUS
SWEEP ANGLE + = CCW SWEEP - = CW SWEEP
0°
180°
RADIUS
+ START ANGLE
90°

EXPLANATION The EW instruction outlines a wedge using the current pen and line type. The arc wedge is referenced to the current pen position which should be thought of as the center of the circle.

The radius defines the size of the circle and can be a positive or negative number in integer or scaled decimal format between -32 768.0000 and 32 767.9999. If scaling is off, the radius is in the plotter units. If scaling is on, the radius is in X-axis user-units. The sign of the radius defines the zero degree reference point for the start angle and sweep angle.

The start angle is in integer format and defines where the first radius is drawn. A positive start angle positions the radius counterclockwise (CCW) from the zero degree reference point; a negative start angle positions the radius clockwise (CW) from the zero degree reference point.

The sweep angle is in integer format between -32 768 and 32 767. The sweep angle defines the number of degrees through which the angle is drawn. A positive sweep angle draws the arc segment CCW; a negative sweep angle draws the arc segment CW. If a sweep angle greater than ±360 degrees is specified, then a 360-degree angle is used.

The chord angle parameter is in integer format between 1-120 and governs the smoothness of the edge. For additional information on the chord angle parameter, see The Circle Instruction, CI, in this chapter.

At the completion of the wedge, the pen is returned to the original position and the pen state is restored.

The following list summarizes your EW options:

Parameter	Type	Range	Default
radius	integer/ decimal	−32 768.0000 to +32 767.9999	none
start angle	integer	MOD 360	none
sweep angle	integer	−32 768 to +32 767	none
chord angle	integer	1-120	5°

An EW instruction with no parameters is not executed but no error is set. Specifying out-of-range parameters sets error 3 and the instruction is ignored. If you send too few parameters, error 2 is set and the instruction is not executed. If you send too many parameters, error 2 is set, the instruction is executed with the first four parameters, and the rest of the parameters are ignored.

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 instruction 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 instructions requires parameters; numeric parameters are ignored. The terminator should be 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 instructions, TL. If no tick length is specified, the length defaults to 0.5% of ($P2_x$ - $P2_y$) 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 Tick Length Instruction, TL

DESCRIPTION The tick length instruction, TL, specifies the length of the tick marks drawn by PACPLOT. The tick length 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 the length of both positive and negative portions of tick marks. The instruction can be used with only one parameter to suspress 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 drawn along the Y-axis, taking P1 as the lower-left corner.

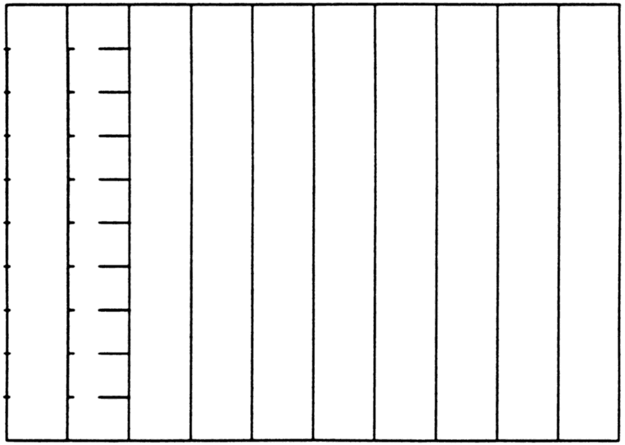
The down and left tick length,tn, determines the length of the downward portion of the tick marks drawn along the X-axis and the left-side portion of the tick marks drawn along the Y-axis, taking P1 as the lower-left corner.

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.

PACPLOT, when initialized, automatically sets the tick length values to 0.5% of the scaling length ($P2_y$-$P1_y$) 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)

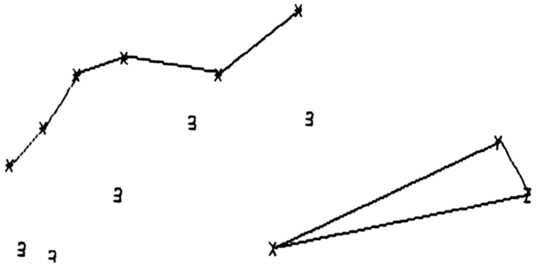
EXPLANATION An SM command without parameters turns off symbol mode. When a parameter is present, it is limited to a single character, which must be one of the printing characters of the character set currently selected.

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.



```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SC0,4000,0,2500;SI.3,.6;SP1;SM*;PA200,1000;"
30 PRINT "PDPA400,1230,600,1560,900,1670,1500,1600,2000,2000;"
40 PRINT "PU;SM;PA100,300;SM3;"
50 PRINT "PA300,500,500,450,900,850,1350,1300,2100,1350PU;"
60 PRINT "SM;PA1900,560;PD;SMY;PA3300,1250;"
70 PRINT "SMZ;PA3500,950;SMX;PA1900,560;PU;SP0;"
```

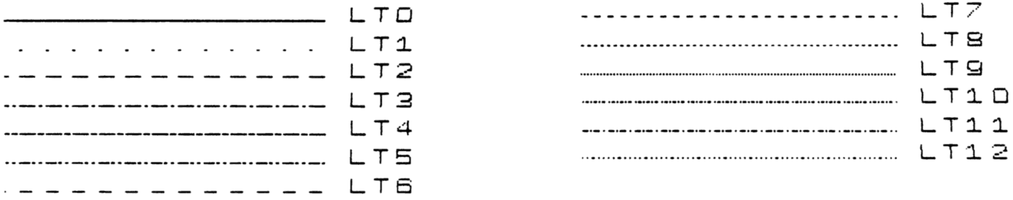
The Line Type Instruction, LT

DESCIPTION 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 12; A parameter in the range 13 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.
The second parameter pattern length is ignored by PACPLOT. The pattern length is fixed to 16 dots on the screen. That equals 256 plotter units.
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, i.e. plot a dot on the screen 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.

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 PACPLOT 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 equivalent3 = CHR$(3) as a BASIC function), 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.

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

```
          CHARACTER SET 0
 ! "#$%&'()*+, -./0123456789:;<=>?@
 ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
 abcdefghijklmnopqrstuvwxyz{|}~
```

Shown next are the symbols which vary from set to set. PACPLOT 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.

The Designate Standard Character Set Instruction, CS

DESCRIPTION The designate standard character set instruction, CS, provides the means of designating one of the two character sets (0 or 33) as the standard character set. Character set 0 is standard ascii set, 33 is the German 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 or 33. 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-out (decimal equivalent 15= CHR$(15) in BASIC) in a label string.. Character set 0 is automatically designated as the standard character set whenever PACPLOT is initialized or set to default values.
A CS command executed while the standard set is selected will immediatly 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 (error3), 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 designatating one of the two character sets as the alternate character set.

USES The instruction can be used 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 or 33. The set designated by 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 = CHR$(14) in BASIC) in a label string. Character set 0 is automatically designated as the alternate character set whenever PACPLOT 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 commend 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 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 autmatically selected when PACPLOT 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 chatacters 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 enterpreted 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 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 Define Terminator Instruction, DT

DESCRIPTOIN 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 be terminated by sending a label terminator at the end of the label character string. ASCII control characters (decimal eqivalent 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.

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 Label Instruction, LB

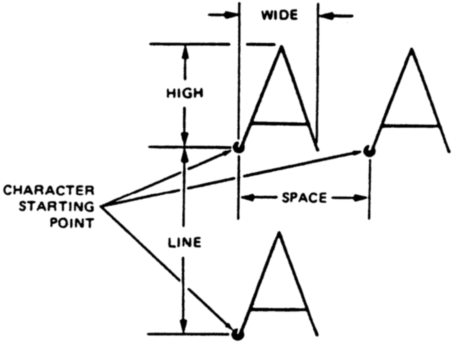
DESCRIPTION The label instruction, LB, provides the means to letter text, expressions, or string variables using the currently defined character set.

USES The label instruction can be used to annotated 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 mnemonis 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. 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) . 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 PACPLOT 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.

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 widths. 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 beween variables.

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 independant of P1,P2 settings. It is especially useful for labeling an Y-axis or labeling a vertical graph.

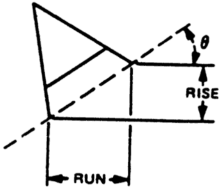
SYNTAX

```
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)

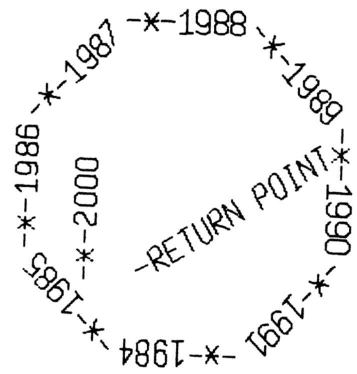
At least one parameter must be effectively nonzero, i.e. |≥0.0004|.
A DI command with a rise parameter of zero will produce horizontal labeling . A DI command with a run parameter of zero will produce vertical labeling.

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 effect the direction of labeling. A DI command in effect until another DI,DR, IN, or DF command is executed.

A DI command updates the carriage return point to the current pen position.
When the angle ∂ ,necessary to establish the desired label direction is known, the command DI cos∂, sin∂ can be used to establish label direction.

The following example labels the year 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 returne point can be clearly seen by observing the pen`s positon 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.



```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SP1;PA2000,3000;SI.5 ,1 "
30 PRINT "DI0,1;LB-*-1984";CHR$(3);"DI1,1;LB-*-1985";CHR$(3)
40 PRINT "DI1,0;LB-*-1986";CHR$(3);"DI1,-1;LB-*-1987";CHR$(3)
50 PRINT "DI0,-1;LB-*-1988";CHR$(3);"DI-1,-1;LB-*-1989";CHR$(3)
60 PRINT "DI-1,0;LB-*-1990";CHR$(3);"DI-1,1;LB-*-1991";CHR$(3)
70 PRINT "PA3000,5250;DI";COS(0);",";SIN(0);"LB-*-2000";CHR$(13);CHR$(10);CHR$(3)
80 PRINT "DI";COS(-45),SIN(-45);"LB-RETURN POINT";CHR$(3)
```

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.

THE RELATIVE DIRECTION INSTRUCTION, DR

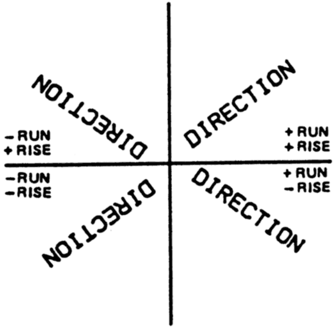
DESCRIPTION The relative direction instruction, DR, specifies the direction in which characters are lettered.

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

SYNTAX DR run, rise terminator
or
DR terminator

EPLANATION Run and rise are in decimal format, 0 to ± 127.9999, and specify a percentage of the algebraic distance P1 and P2 where run is the desired percentage (-128 to 127.9999) of $P2_x - P1_x$, rise 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 determine 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 is executed.

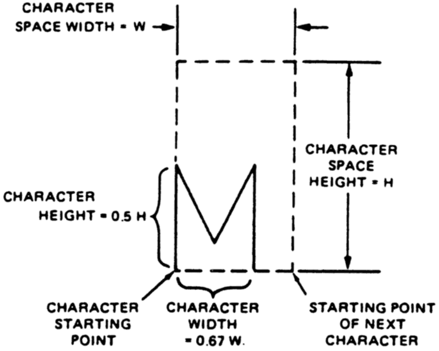
A DR command with no parameters will default to the values DR 1 , 0 (horizontal).

Specifying both parameters as zero will set error 3, and having only one or more than two parameters will set error 2. PACPLOT 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 1 $^1/_2$ 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.


CHARACTER SPACE WIDTH = W
CHARACTER SPACE HEIGHT = H
CHARACTER HEIGHT = 0.5 H
CHARACTER STARTING POINT
CHARACTER WIDTH = 0.67 W.
STARTING POINT OF NEXT CHARACTER

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

THE CHARACTER PLOT INSTRUCTION, CP

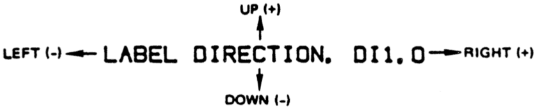
DESCRIPTION The character plot instruction, CP, moves the pen the specified number of character-space fields.

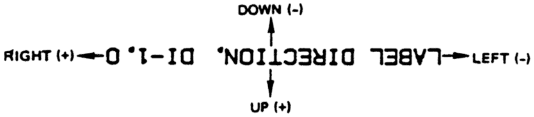
USES The instruction can be used to move the pen any number of character spaces or lines from a point on the plotting surface, to align with a left-hand margin, or to center or right-justify a label. Thus, the label can be moved slightly above or below a line, space or lines can be inserted in text, 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 sommand performs a carriage return and line feed, moving one character-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 the pen position at the last DI or DR command. Refer to The Label Instruction in this chapter.

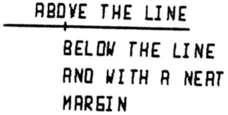
When parameters are specified, the CP comand moves the pen the specified number of character-space-field widths to the right (a positive value) or the left (a negative value). Note that right, left, up, and down are relative to the label direction, where a positive value means from P1 toward P2. This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



```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SI.3,.6  ;SP1;PA1000,2000PDPR3000,0PU;PR-3000,0;"
30 PRINT "CP3,.35;LBABOVE THE LINE";CHR$(3);"PA2000,2000;"
40 PRINT "XT;CP0,-.95;LBBELOW THE LINE"
50 PRINT "AND WITH A NEAT";
60 PRINT CHR$(3);"CP;LBMARGIN";CHR$(3);
```

THE ABSOLUTE CHARACTER SIZE INSTRUCTION, SI

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

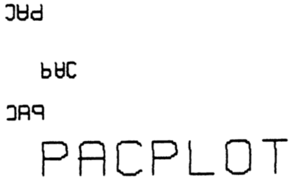
USES The instruction can be used to change the character size from its default value or another value and establish absolute character sizing in centimetres so character size is not dependent on the settings of P1 and P2.

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 PACPLOT 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 examples letter PACPLOT at the specified width of 1 cm and height of 1.5 cm.
Negative SI parameters will produce mirror images of labels. A negative SI width parameter will mirror labels in the right-to-left direction.
A negative height parameter will mirror labels in the top-to-bottom direction.
Two negative SI parameters will mirror the label in both directions and the label will appear to be rotated 180 degrees.



```
5 PRINT "IN;PA2000,1000;"
10 PRINT "SI1,1.5;LBPACPLOT";CHR$(3)
15 PRINT "PA2000,2000;"
20 PRINT "SI-.35,.6;LBPAC";CHR$(3)
25 PRINT "PA2000,3000;"
30 PRINT "SI.35,-.6;LBPAC";CHR$(3)
35 PRINT "PA2000,4000;"
40 PRINT "SI-.35,-.6;LBPAC";CHR$(3)
50 PRINT "DF;SI1.3,1.8;PA3000,6000;"
60 PRINT "SL1;LBPAC";CHR$(3)
70 PRINT "SL-1;PR1300,0;LBPAC";CHR$(3)
```

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 alow 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.

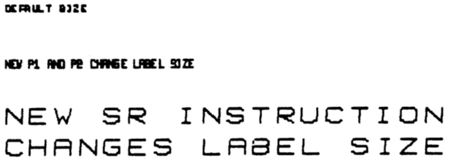
SYNTAX

```
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, produces letters the same size as an SI command without parameters.

An SR command shows in effect until another valid SI or SR command is executed or PACPLOT 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 9000-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.



DEFAULT SIZE

NEW P1 AND P2 CHANGE LABEL SIZE

NEW SR INSTRUCTION
CHANGES LABEL SIZE

```
10 PRINT "IN;SP1;PA1000,6000;LBDEFAULT SIZE";CHR$(3)
20 PRINT "IP0000,0000,9000,9000;DIPA1000,5000;"
30 PRINT "LBNEW P1 AND P2 CHANGE LABEL SIZE";CHR$(3);"SR3,3;"
40 PRINT "PA1000,4000;LBNEW SR INSTRUCTION"
50 PRINT "CHANGES LABEL SIZE";CHR$(3)
```

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.

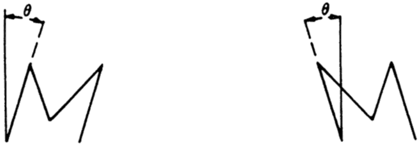
THE CHARACTER SLANT INSTRUCTION, SL

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 ø (terminator)
or
SL (terminator)

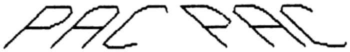
EXPLANATION The instruction may be used with or without parameters. When parameters are included, the first parameter is interpreted as the tangent of the angle from vertical as shown below. Parameters following the first parameter are ignored. An SL command without parameters defaults to the same value as SL 0 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 .

The following example letters PAC at a slant of +45 degrees and -45 degrees. Line 50 and 60 in the example of the SI command result in this picture.



THE USER DEFINED CHARACTER INSTRUCTION, UC

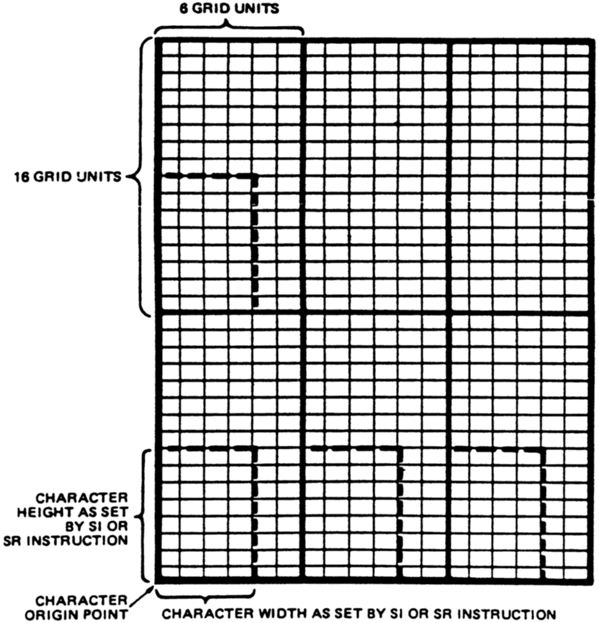
DESRIPTION The user defined character instuction,UC, provides the means to draw characters of your own design. It is included in the instruction set of the 7475 plotter with an HP-IL interface.

USES This instruction can be used to create symbols not included in PACPLOT`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
Each segment of the characters 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 fields and thus the grid is always twice the current character height and 1 1/2 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.


6 GRID UNITS
16 GRID UNITS
CHARACTER HEIGHT AS SET BY SI OR SR INSTRUCTION
CHARACTER ORIGIN POINT
CHARACTER WIDTH AS SET BY SI OR SR INSTRUCTION

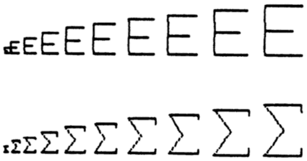
Character Grid

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- and 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 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 is complete, the pen returne to its up/down status as set by PU or PD.

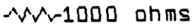
The position of the pen when the UC command is executed becomes the character origin point. The initial X,Y increment is relative to the character origin point and each subsequent move is relative to the last commanded pen position. Upon completion of the user defined character, the pen is automatically 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 upercase 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.



```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SP2;PA1000,1000;"
30 FOR A=.20 TO 1.6 STEP .15
40 PRINT "SI",A;",";A*1.4
50 PRINT "UC4,7,99,0,1,-4,0,2,-4,-2,-4,4,0,0,1;"
60 NEXT A
70 PRINT "PA1000,2750;"
80 FOR B=.20 TO 1.6 STEP .15
90 PRINT "SI",B;",";B*1.4
100 PRINT "LBE";CHR$(3)
110 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.



```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SP1;PA1000,5000;SI.4 ,.6"
30 PRINT "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;"
40 PRINT "CP3.25,0;LB1000 ohms";CHR$(3)
```

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. With this Feature it is very easy to enlarge or stretch a user defined character in x or y position.

PARAMETER INTERACTION IN LABELING COMMANDS

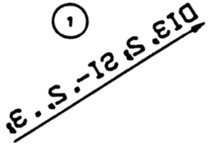
There are three factors which interact and affect the direction and mirroring of labels; the label direction as specified by DI or DR commands or default direction, the sign of the parameters for the size commands SI or SR, and the relative position of P1 and P2. These interactions are complex. This 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.

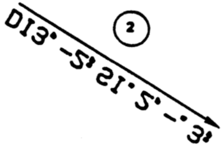
The labels used in the illustration are the commands which cause the direction, size, and mirroring of the label. All descriptions are in terms of the standard X, Y coordinate system. An arrow is shown for each label; this arrow is the baseline along which labeling occurs and shows the left-to-right 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 illustration, 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 0 and 10 224; the values used for the Y-coordinates of P1 and P2 are 0 and 7968 or 7296 with a mouse connected.

USE OF DI AND SI

When DI and SI commands are used together, the DI command establishes the label`s direction and the SI 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.

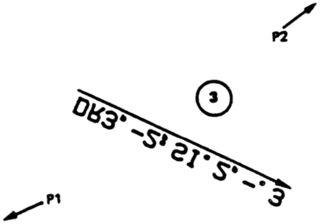
Two examples of mirrored labels are shown below. In the first example, the DI parameters 3, 2 place the directional line in the first quadrant. The negative width parameter of the SI command mirrors the label in the right-to-left direction. In the second example, the DI parameters 3,-2 place the directional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



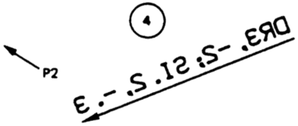


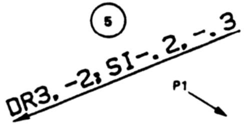
USE OF DR AND SI

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 DR 3,-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.

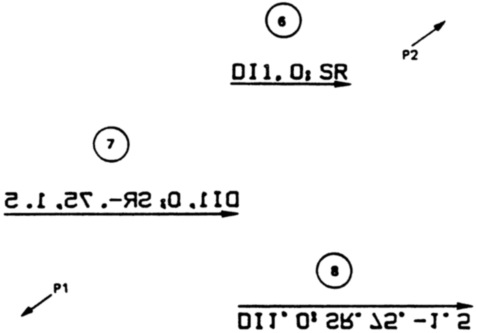




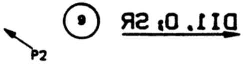
USE OF DI AND SR

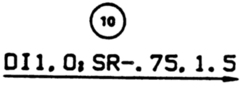
When the DI command is used with SR, only the DI command affects the directional baseline of labels; changes in the relative positions of P1 and P2 do not affect the baseline. Mirroring about this baseline will occur when either a negative SR width or height parameter with a positive difference ($P2_x$ - $P1_x$) or ($P2_y$ - $P1_y$) or a positive SR parameter and a negative difference are present. If respective parameters and differences are both negative, no mirroring will occur.

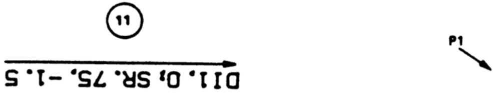
Label direction is horizontal for all illustrations in this section. The first three illustrations are drawn with P1 and P2 at their power-on settings. In example 6, the SR ; command is the same as SR. 75,1.5. Since the parameters are positive, there is no mirroring. In example 7, the negative width parameter causes mirroring right-to-left. In example 8, the negative height parameter causes mirroring top-to-bottom.



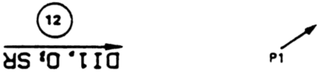
In the next three illustrations, P1 and P2 have changed so P1 is lower right and P2 is upper left. Hence ($P2_x$ - $P1_x$) is negative and anything with a positive SR width parameter is mirrored right-to-left, e.g., illustrations 9 and 11. The effect of the negative width parameter in illustration 10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

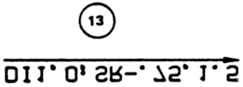


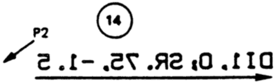




In the next illustrations, P1 and P2 have both been flipped so P1 is upper right and P2 is lower left. Now any positive parameter causes mirroring and any negative parameter cancels mirroring. This can be seen in examples 12,13, and 14.







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 algebraic differences ($P2_x$ - $P1_x$) and ($P2_y$ - $P1_y$) to establish label direction, and SR parameters interact with these differences to create mirroring. Signs of both parameters and 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.

OBTAINING INFORMATION FROM PACPLOT

WHAT YOU'LL LEARN IN THIS CHAPTER

Up to this time we have mainly been concerned with sending information or data to PACPLOT. Sometimes, however, we want to know something about PACPLOT, its current pen position, its status, whether an error has occured, or what capabilities PACPLOT has. In this chapter you will learn about most of PACPLOT's output instructions. The output P1 and P2 and output window instructions are discussed elsewhere. All other output instructions are discussed in this chapter. 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 PACPLOT at the end of a plotter response to an output command. The two characters, carriage return and line feed, are the output terminator.

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

With mouse

A3	A4
0 ≤ X ≤ 14 459	0 ≤ X ≤ 10 224
0 ≤ Y ≤ 10 318	0 ≤ Y ≤ 7 296

Without mouse

A3	A4
0 ≤ X ≤ 14 459	0 ≤ X ≤ 10 224
0 ≤ Y ≤ 11 268	0 ≤ Y ≤ 7 968

No positive sign is output.

The Output Commanded Position and Pen Status Instruction, OC

DESCRIPTION The output commanded position and pen status (up or down) associated with the last valid pen position command.

USED 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 infomation 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 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: 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.

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 PACPLOT.

SYNTAX OE (terminator)

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

When an OE command is received, PACPLOT 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	Not used

[You should note that anytime PACPLOT receives an unpaired alphabetic character, errror 1 will be set. Thus, an analphabetic 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 PACPLOT 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.

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. This Factor is not true with PACPLOT and a video screen but is used for compatibility.

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.

PACPLOT will always output the following character string:

APS V2.0 as 7475A [TERM]

[**TERM**] is the output terminator .

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 enviroment 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 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 interal 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 ouput of OP
4	2	Digitized point available; cleared by reading digitized value
8	3	Initialized; cleared by reading OS .Ready for data;
32	5	Error; cleared by executing OE
64	6	Not used (always 0)
128	7	data send by output commands ; cleared by OS or IN

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

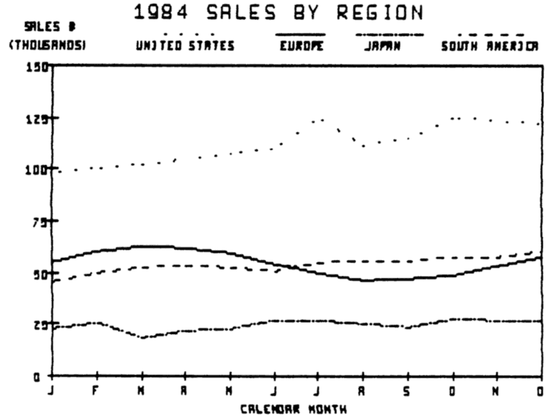
The following table shows the number and type of items in the response to each HP-GL output command. The table includes output commands explained in Chapter 2 and 6 as in this chapter. This table will be helpful when programming in languages such as FORTRAN which require you to specify the type of and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

Instruction	Number of Parameters Returned *	Type and Range
OA	3	integers, all ≤ 5 digits
OC	3	decimals, all ≤ 11 digits
OD	3	integers, all ≤ 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 ≤ 5 digits; 2nd and 4th ≤ 4 digits
OS	1	integer, ≤ 3 digits
OW	4	integers, 1st and 3rd ≤ 5 digits; 2nd and 4th ≤ 4 digits

PUTTING THE COMMANDS TO WORK

WHAT YOU'LL LEARN IN 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 concept 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 if 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 plotter is important, a semicolon or a line feed should always be used as the terminator and parameters should be separated by commas.

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 PACPLOT 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 units 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 150000 - the actual range of sales dollars. There are two reasons for this. First, the largest number accepted by PACPLOT is 32767; 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 (150000). 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: line 10 - 40.

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 TL1.5,0. 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 of this manual. 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.

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.) Sencondly, 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: lines 50-110 and line 310.

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 are : lines 120-190.

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.

After drawing the line, the pen moves to the legend area below the graph title and draw 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. 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: lines 230 -280.

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 PACPLOT 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 PACPLOT 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: lines 290-380.

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.

A complete listing of the program follows. This listing contains all the BASIC statements necessary to have this program run on an HP-71 computer with an HP-IL interface and PACPLOT. In some PRINT statements, semicolons or commas are used to ensure that HP-IL commands will have the necessary separators or no extra spaces. You may need to make changes for your computer´s BASIC, or you use some other programming language and send the strings of HP-GL commands using your language´s output statements and looping techniques.

LISTING OF LINEGRAPH PROGRAM

```
10 PRINTER IS 'PACPLOT'
20 PRINT "IN;SP1;IP1250,750,9250,6250;"
30 PRINT "DI;SC1,12,0,150;"
40 PRINT "PU1,0PD12,0,12,150,1,150,1,0PU"
50 PRINT "SI.21,.4;"
60 FOR X=1 TO 12
70 PRINT "PA";X,",0;XT;"
80 READ A$
90 PRINT "CP-.33,-1;LB";A$;CHR$(3)
100 NEXT X
110 PRINT "PA6.5,0;CP-7,-2.0;LBCALENDAR MONTH";CHR$(3)
120 FOR Y=0 TO 150 STEP 25
130 PRINT "PA1,",Y,"YT;"
140 IF Y<100 THEN PRINT "CP-3,-.25;LB";Y;CHR$(3)
150 IF Y>99 THEN PRINT "CP-4,-.25;LB";Y;CHR$(3)
160 NEXT Y
170 PRINT "PA1,150;CP-3.5,2 LBSALES $";CHR$(3);"CP-9,-1"
180 PRINT "LB(THOUSANDS)       UNITED STATES       ";CHR$(3)
190 PRINT "LBEUROPE     JAPAN     SOUTH AMERICA  ";CHR$(3)
210 PRINT "SP2;PA6,150 SI.4,.6 CP-9.5,1.8"
220 PRINT "LB1984 SALES BY REGION";CHR$(3)
230 PRINT "SP3;LT3,6;PA1,23PD2,25,3,18,4,22,5,23"
240 PRINT "PD6,27,7,27,8,25,9,24,10,28,11,27,12,27PU"
250 PRINT "PA7.8,165 PD9.3,165 PU"
260 PRINT "SP4;LT6,8;PA1,45;PD;PA2,50,3,52,4,53,5,52"
270 PRINT "PD6,51,7,55,8,56,9,56,10,58,11,58,12,60PU"
280 PRINT "PA10.1,165 PD11.6,165 PU"
290 PRINT "LT;SP5"
300 FOR X=1 TO 12
310 READ Y
320 PRINT "PA";X;Y;"PD"
330 NEXT X
340 PRINT "PU6,165PD7.1,165PU"
350 PRINT "SP6;LT1,3"
360 GOSUB 1000
370 DATA "J","F","M","A","M","J","J","A","S","O","N","D"
380 DATA 55,60,63,62,59,54,50,46,47,49,53,58
390 STOP
1000 REM 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 "LT1,3 PU3.2,165 PD4.7,165SP0;"
1100 RETURN
1110 END
```

BAR GRAPHS - FILLING AND HATCHING

Two kinds of area fill are commonly used in bar graphs ; 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.

PRODUCING A BAR GRAPH

Scaling the Axes

In the following bar graph titled "Sales Volume by Region", we are plotting sales over a three-year period. For readability, the X-axis is scaled to provide a comfortable margin of space before and after each bar. The Y-axis is scaled from 0 to 500 to represent sales in thousands of dollars.

Plotting the Title

The title and axes are drawn with a wide pen (stored in stall two) for emphasis. The title is drawn first, with characters that are little more than twice the default size. All labels are centered and offset slightly from the data area with the CP instruction.

Labeling the Axes

The bars on the X-axis are labeled without tick marks, using a narrow pen and characters that are slightly larger than default size. The Y-axis is labeled with tick marks and an extra label to show the scaling used (K$).

Labeling the Bar Segments

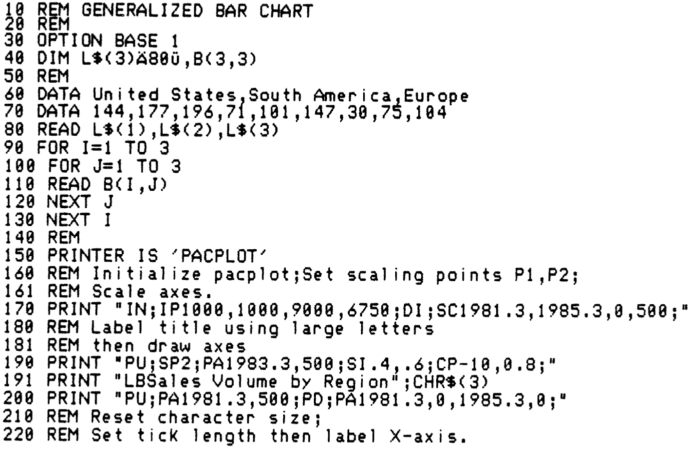
The data for each bar is input using read and data statements. This approach allows easy modification of the label data. Each segment label is centered next to the rightmost bar by computing a Y-axis position that is equal to the height of the prior segments plus one-half the height of the current segment.

Filling and Edging Each Segment

The data for each bar segment is stored in a three by three array. Each array element contains the height of a segment with respect to the Y-axis scaling. The bars are drawn from bottom to top using the FT, RA, and EA instructions to define, fill, and edge each stacked rectangular segment. Wide pens are used in stalls 3, 4, and 5 of the carousel for filling the bars.

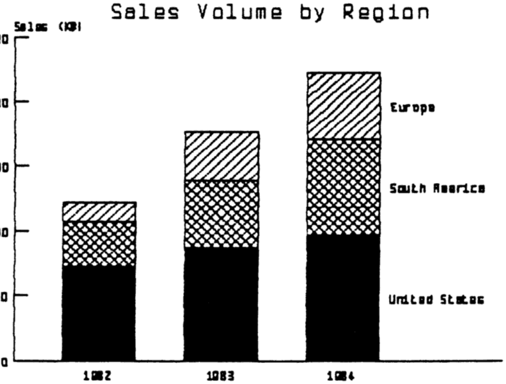
Completion of Bar Graph

At the completion of the program, the scaling points are reset to their default location, the pen is raised and put away, and the finished plot is presented for viewing.



```
10 REM GENERALIZED BAR CHART
20 REM
30 OPTION BASE 1
40 DIM L$(3)Ä80ü,B(3,3)
50 REM
60 DATA United States,South America,Europe
70 DATA 144,177,196,71,101,147,30,75,104
80 READ L$(1),L$(2),L$(3)
90 FOR I=1 TO 3
100 FOR J=1 TO 3
110 READ B(I,J)
120 NEXT J
130 NEXT I
140 REM
150 PRINTER IS 'PACPLOT'
160 REM Initialize pacplot;Set scaling points P1,P2;
161 REM Scale axes.
170 PRINT "IN;IP1000,1000,9000,6750;DI;SC1981.3,1985.3,0,500;"
180 REM Label title using large letters
181 REM then draw axes
190 PRINT "PU;SP2;PA1983.3,500;SI.4,.6;CP-10,0.8;"
191 PRINT "LBSales Volume by Region";CHR$(3)
200 PRINT "PU;PA1981.3,500;PD;PA1981.3,0,1985.3,0;"
210 REM Reset character size;
220 REM Set tick length then label X-axis.
```

```
221 REM
230 PRINT "PU;SP1;SI.2,.4;TL1.5,0;"
240 FOR X=1982 TO 1984
250 PRINT USING "K";"PU;PA",X,",0;CP-1.83,-1;LB",X,CHR$(3)
260 NEXT X
270 REM Add ticks and labels to Y-axis
280 FOR Y=0 TO 500 STEP 100
290 PRINT "PU;PA1981.3,";Y;"YT;"
300 IF Y=0 THEN PRINT "CP-2.5,-.25;"
310 IF Y<>0 THEN PRINT "CP-4.5,-.25;"
320 PRINT "LB";Y;CHR$(3)
330 NEXT Y
340 REM
350 PRINT "PU;PA1981.3,500;CP0,.5;LBSales (K$)";CHR$(3)
360 REM Center segment labels using prior height +1/2
370 REM current height
380 FOR I=1 TO 3
390 Y=0
400 FOR J=1 TO I-1
410 Y=Y+B(J,3)
420 NEXT J
430 Y=Y+B(I,3)/2
440 PRINT USING "K";"PU;PA1984.4,",Y
441 PRINT USING "K";"CP0,-.25;LB";L$(I);CHR$(3)
450 NEXT I
460 REM Draw and fill each bar
470 FOR I=1 TO 3
480 PRINT "SP";I+2;"PT.7;"
490 K=1
500 FOR X=1982 TO 1984
510 Y1=0
520 REM Compute Y-axis start point for each bar segment
530 REM
540 FOR J=1 TO I-1
550 Y1=Y1+B(J,K)
560 NEXT J
570 REM Compute Y-axis end point for each segment
580 Y2=Y1+B(J,K)
590 K=K+1
600 REM Select fill type
610 IF I=1 THEN PRINT "FT1;"
620 IF I=2 THEN PRINT "FT4,226,45;"
630 IF I=3 THEN PRINT "FT3,226,45;"
640 REM Move to start point for segment
650 REM then fill defined bar area
660 PRINT "PU;PA";X-.3;Y1
670 PRINT "RA";X+.3;Y2
680 REM Return to segment start point and outline segment
690 PRINT "PU;PA";X-.3;Y1
700 PRINT "EA";X+.3;Y2
710 NEXT X
720 NEXT I
730 REM Put the pen away and present paper for viewing
740 PRINT "IP;PA1976.3,0;SP0;"
750 END
```


Sales Volume by Region
Sales (K$)
500
400
300
200
100
0
1982
1983
1984
Europe
South America
United States

APS 2.0 specific extensions to HP-GL

The commands described in this section are Advanced PAC Screen V.2.0 specific commands. They are Extensions to the HP-GL and not part of the standard HP-Graphics Language command set.
The commands covered here are very helpful in debugging plotter software, printing the graphic screen to a Centronics or Thinkjet printer a.s.o. You should always realize, that programs using these commands will not run on a HP Plotter without modifications.

Debugging Plotter Software with the Advanced Pac Screen

Locating bugs with the Trace Function
The Trace Function described below is very helpful to find bugs in plotter software or to learn how the HPGL works. When writing Your own plot programs You will find it a lot easier with the plot status at hand using the TR1 mode. If You think the program runs correctly You still can switch on TR2. No status line appears on the screen until an error is encountered.

Learning the HP Graphics Language, HPGL with the Advanced Pac Screen 2.0
Advanced Pac Screen 2.0 offers You another powerful debugging tool with its large text memory within PACTEXT. You easily can send your Plotting data not only to PACPLOT but also to PACTEXT. With the mouse you can scroll the 125 lines of HPGL commands up and down to locate errors or just to learn how the HPGL works. If You want trace more than 125 lines You might use PACCEN or the HP THINKJET as second output device.

The Trace Mode Instruction, TR

DESCRIPTION The trace mode instruction was added to the standard HP-GL command set to support debugging in plotter input strings. Therefore some important informations about the current plotter status are displayed in a status line displayed inverse and on top of the screen of the APS 2.0. Its appearance and contents are explained below. The data is always related to the last HP-GL instruction which triggered the status line.

Structure of the Trace Line :

*CMDNO CMDID ERROR plot:**DIR, UNITS** pen mode: **STATE,MODE** at: **X,Y**

CMDNO instant number of last command which triggered the Trace Function (range from 1 to 2999)

CMDID name of command which triggered the Trace Function.

ERROR error number caused by the command . Possible error codes are

No error
Error 1 Instruction not recognized. Advance Pac Screen has received an illegal character
sequence
Error 2 Wrong number of parameters. Too many or too few parameters
Error 3 Bad parameter. Parameter sent with an instruction are out of range for that instruction
Error 4 Not used
Error 5 Unknown character set. A character set out of 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.

Error 7 Not used
Error 8 Not implemented. Vector received while wheel raised

DIR - ABSOLUTE or RELATIVE coordinates (see PA or PR instruction)

UNITS - USER or Plotter units (see SC instruction)

STATE - pen UP or pen DOWN

MODE - writing modes of pen are
ADD,
REPlace
INVert or
ERAse

X,Y - last pen position;always in plotter units

SYNTAX TR (mode;)

EXPLANATION Omitting parameters causes deactivation of the trace Status line is removed from the screen and the Trace Function is not active.

mode = 0 trace mode will be deactivated and status line removed

= 1 general trace conditon. Status line is updated as soon finishs a GL command.

= 2 trace only if an error has been discovered. The status till next error is encoutered or trace mode is changed.

The instruction count is reset to 1 whenever the trace mode is modified, but remains active. In case of out of range error for the TR instruction the status display will be deactivated.

Printing Graphic with the Advanced Pac Screen 2.0

The following commands desribe how to print graphics with the Advanced Pac Screen 2.0 sourcing data and sending it to various printers. There are a few things You should keep in mind.

Caution

1. Whenever You specify the mouse as an input device to digitize a point or select a range PACPLOT waits till the mouse button is pressed and released again. The Loop stops until this condition is fulfilled. If the Text page is on top of the screen You may not even notice that PACPLOT waits for mouse input. Advanced Pac Screen 2.0 is blocked totally.

2. Whenever You make Advanced Pac Screen 2.0 send Data on the Loop You have to collect them at once. In case You miss it after an OD command, only the data will be lost.
If You produce more output with a HL or LO command the internal buffer of PACPLOT will overflow and PACPLOT will stop Loop operation till You collect your data. The two other devices PACCEN and PACTEXT will work as usual.

The Hardcopy to Centronics Instruction, HC

Description The hardcopy to centronics instruction is used to bitmap copy of the specified area to the
Advanced Pac Screen printer port.

Syntax HC(mode,$X_{lower-left}$,$Y_{lower-left}$,$X_{upper-right}$,$Y_{upper-right}$;)

Explanation If parameters are omitted, the total Graphic screen will be dumped. The -mode- parameter selects one out of three possibilities to select the window around the section to be printed . The selected section of the screen allways is printed beginning at the left side of the paper. You can alter this by moving the left margin of Your printer further to the right side.

mode =0 the window encloses the whole screen.

mode =1 The window will be selected with the connected mouse. A cross hair cursor apears on the graphic screen. By moving the mouse on the table You can select the lower left corner. When You have found the right position press the mouse button and keep it pressed. The cross hair cursor disappears from the screen. Now move the mouse. A box appears on the screen which follows any movement You make on the table. The contents of this box including the borders of the box will be printed when You release the mouse button. The HC1 command will be ignored if there is no mouse connected to APS 2.0.

mode =2 requires that two corners of the window are specified. If scaling is on, values for $X_{lower-left}$, $Y_{lower-left}$, $X_{upper-right}$ and $Y_{upper-right}$ are interpreted as user units. Otherwise they are interpreted as plotter units in integer format .

All parameters must be integers between -32768 to 32767. Specifying parameters out of range error 3 is set and the command will be ignored.

The command is aborted if

$X_{lower-left} = X_{upper-right}$ or $Y_{lower-left} = Y_{upper-right}$.

i.e. if there is no window defined. The Format of the bitimage graphics data is compatible with EPSON FX, or RX 80 printers. If the default control sequences do not match Your printer You can alter them within APS 2.0. Refer to Chapter PACCEN for detailed information. There You find examples how to adapt Advanced Pac Screen 2.0 to Your printer.

The Hardcopy to Loop in Thinkjet Format Instruction, HT

Description The hardcopy to loop in thinkjet format instruction is used to send a copy of the specified window to the HP Interface Loop using HP Thinkjet controll sequences for printing bit image graphics. For more information about the Thinkjet format refer to the appropriate Thinkjet Reference Manual. The window to be printed is specified by the coordinates of the lower left and upper right corners.

Syntax HT(mode,$X_{lower-left}$,$Y_{lower-left}$,$X_{upper-right}$,$Y_{upper-right}$;)

Explanation If parameters are omitted, the total Graphic screen will be dumped. The -mode- parameter selects one out of three possibilities to select the window around the section to be printed . Printing of the selected section from the screen allways begins at the left side of the paper. You can alter this by moving the left margin of Your printer further to the right side.

mode =0 the window encloses the whole screen.

mode =1 The window will be selected with the connected mouse. A cross hair cursor apears on the graphic screen. By moving the mouse on the table You can select the lower left corner. When You have found the right position press the mouse button and keep it pressed. The cross hair cursor disappears from the screen. Now move the mouse. A box appears on the screen which follows any movement You make on the table. The contents of this box including the borders of the box will be printed when You release the mouse button. The HT1 command will be ignored if there is no mouse connected to the APS 2.0.

mode =2 requires that two corners of the window are specified. If scaling is on, values for $X_{lower-left}$, $Y_{lower-left}$, $X_{upper-right}$ and $Y_{upper-right}$ are interpreted as user units. Otherwise they are interpreted as plotter units in integer format .

All parameters must be integers between -32768 to 32767. Specifying parameters out of range error 3 is set and the command will be ignored.
The command is aborted if

$$X_{lower-left} = X_{upper-right} \text{ or } Y_{lower-left} = Y_{upper-right}.$$

i.e. if there is no window defined. In this case Bit 6 of PACPLOT's Status Byte will be set. The HPIL Interface protocol provides no means to let control PACPLOT the transfer of data over the loop. Therefore it is Your task to collect the data and send them to the Thinkjet. To make it as easy as possible PACPLOT informs You how many blocks of 256 Bytes You have to transfer over the loop. That makes it easy to produce graphic dumps of any size by using a subroutine which is always called after each HT command. A subroutine which can fullfil this task is listed below. The subroutines LOGOOUT, LOGOIN and PLOT are provided at the end of this Chapter.

```
10 PRINTER IS :PACPLOT !      The semicolon  at the end of the PRINT command is
20 PRINT "IN;PP;HT1;"; !      needed to suppress CR LF. Otherwise the hardcopy
30 !                          will not work!!!
40 !
50 FOR A=1 TO 10 !            Test the plotter's Status Byte to see if there is
60 IF BIT(SPOLL("PACPLOT"),6)=1 THEN 100 ! data available. Do it ten times
70 NEXT A !                               because the plotter needs a little
80 DISP "NO DATA AVAILABLE" @ STOP !      time.
90 !
100 CALL PLOT !        Now call the PLOT subroutine to transfer the hardcopy
110 DISP "READY" !     data from APS to the Thinkjet.
```

To produce Your own letter head or bussines label You also could store the graphic data on tape or diskette using the mouse to select the appropriate section of the screen.
Anytime You want to print it afterwards You just transfer the data from the tape or diskette to the thinkjet. All necessary control sequences to switch the Thinkjet to graphics mode, print the graphic and switch back to text mode are included in the data strings delivered by APS 2.0. Two programs to do this is listed below.

```
10 !        Call the LOGOIN subroutine to transfer the data from data file
20 !        to your Thinkjet.
30 !        Remember that line 230 in the LOGOIN subroutine has to be
40 !        changed into PRINTER IS :HP2225B to direct the data to the
50 !        Thinkjet but not to PACPLOT device.
60 !
70 CALL LOGOIN
80 DISP "READY"
```

```
10 PRINTER IS :PACPLOT
20 !
30 !       The semicolon  at the end of the PRINT command is needed to
40 !       suppress CR LF. Otherwise the hardcopy will not work!!!
50 !
60 PRINT "IN;PP;HT1;";
70 !
80 !       Test the plotter´s Status Byte to see if there is data available.
90 !       Do it ten times because the plotter needs a little time.
100 !
110 FOR A=1 TO 10
120 IF BIT(SPOLL("PACPLOT"),6)=1 THEN 190
130 NEXT A
140 DISP "NO DATA AVAILABLE" @ STOP
150 !
160 !       Call the LOGOOUT subroutine to transfer the hardcopy data
170 !       from APS to a data file.
180 !
190 CALL LOGOOUT
200 DISP "READY"
```

The Define Logo and Output to Loop Instruction, LO

Description The define logo and output to loop instruction may be used to send a copy of the specified plotter area to the HP Interface Loop. Under control of a user defined program these data may be saved on any mass storage device for later use by the LI command.
With help of these commands it is possible to copy sections from and to the screen. You could shift sections of the screen around using the RAM of Your computer as temporary buffer and the mouse to specify size and place. The window to be copied is specified by the coordinates of the lower left and upper right corners.

Syntax LO(mode,$X_{lower-left}$,$Y_{lower-left}$,$X_{upper-right}$,$Y_{upper-right}$;)

Explanation If parameters are omitted, the total Graphic screen will be dumped. The -mode- parameter selects one out of three possibilities to select the window around the section to be printed . The selected section of the screen allways is printed beginning at the left side of the paper. You can alter this by moving the left margin of Your printer further to the right side.

note: to copy the whole screen You need 23 kByte of memory !!!!!!

mode =0 the window encloses the whole screen.

mode =1 The window will be selected with the connected mouse. A cross hair cursor apears on the graphic screen. By moving the mouse on the table You can select the lower left corner. When You have found the right position press the mouse button and keep it pressed. The cross hair cursor disappears from the screen. Now move the mouse. A box appears on the screen which follows any movement You make on the table. The contents of this box including the borders of the box will be printed when You release the mouse button. The LO1 command will be ignored if there is no mouse connected to the APS 2.0.

mode =2 requires that two corners of the window are specified. If scaling is on, values for $X_{lower-left}$, $Y_{lower-left}$, $X_{upper-right}$ and $Y_{upper-right}$ are interpreted as user units. Otherwise they are interpreted as plotter units in integer format .

All parameters must be integers between -32768 to 32767. Specifying parameters out of range error 3 is set and the command will be ignored.
The command is aborted if

$X_{lower-left} = X_{upper-right}$ or $Y_{lower-left} = Y_{upper-right}$.

i.e. if there is no window defined. In this case Bit 6 of PACPLOT's Status Byte will be set. The HPIL Interface protocol provides no means to let control PACPLOT the transfer of data over the loop. Therefore it is Your task to collect the data and send them to the mass storage device. To make it as easy as possible PACPLOT informs You how many blocks of 256 Bytes You have to transfer over the loop. That makes it easy to produce graphic dumps of any size by using a subroutine which is always called after each LO command..The subroutines LOGOOUT, LOGOIN and PLOT are provided at the end of this Chapter.

To produce Your own letter head or bussines label You also could store the graphic data on tape or diskette using the mouse to select the appropriate section of the screen. Anytime You want to have it on screen afterwards You just transfer the data from the tape or diskette to PACPLOT.The subroutines LOGOOUT, LOGOIN and PLOT are provided at the end of this Chapter. A program to do this is listed below.

```
10 PRINTER IS :PACPLOT !   The semicolon at the end of the PRINT command is
20 PRINT "IN;PP;LO1;"; !   needed to suppress CR LF. Otherwise the transmission
30 !                       will not work!!!
40 !
50 FOR A=1 TO 10 !         Test the plotter's Status Byte to see if there is
60 IF BIT(SPOLL("PACPLOT"),6)=1 THEN 100 ! data available. Do it ten times
70 NEXT A !                                because the plotter needs a little
80 DISP "NO DATA AVAILABLE" @ STOP !       time.
90 !
100 CALL LOGOOUT !       Call the LOGOOUT subroutine to transfer the LOGO data
110 DISP "READY" !       from APS to a data file.

10 PRINT "IN;LI;";
20 CALL LOGOIN !        Call the LOGOIN subroutine to transfer the LOGO data
30 DISP "READY" !       from the data file to APS.
```

The Input Logo from Loop Instruction, LI

Description The logo input from loop instruction is used to copy a section defined earlier with the LO instruction from the mass storage device to the screen. The window defined by the LO instruction replaces the section of Your plot You position it. Advanced Pac Screen 2.0 needs the position of the upper left corner as a Reference point .

Syntax LI(XRef, YRef;)

Explanation Omitting paramaters causes designation of the reference point interactively with the mouse, as described in the DP command. If there is no mouse connected to the Advanced Pac Screen 2.0 the LI command without parameters is aborted and no error is set.

The Digitize Point with Rubber Line Cursor Instruction, DL

Description The digitize point with rubber line cursor instruction is designed to support some special tasks in user applications like CAE/CAD software.
To get coordinates of one point interactively the user may choose between the DP or DL instruction. Instead of the cross-hair cursor for the DP command, Advanced Pac Screen 2.0 will show a rubber-line only, wrapped to the last pen position after execution of any instruction like PU,PD,AA,AR etc. A Rubber line is a line whose one end is fixed to a point. The other end can be moved over the screen with the mouse. The line alters direction and length. This command is especially helpful if you want to plot using the mouse as a pen. You can easily determine the direction and appearance of a line before you actually draw it.

Syntax DL;

Explanation When Advanced Pac Screen has received the DL command and the mouse button is not hold down, a cross-hair cursor is displayed at the location of the last pen position till the mouse button is pressed. Then the cursor changes from a cross-hair to a rubber-line and can be moved around the whole screen until the button is released. The rubber line appears on the screen whether the pen is up or down. The triggered position is stored within the Advanced Pac Screen 2.0 until another digitize instruction has been recognized. It is not possible to digitize more than one point with the DL instruction. If you don´t want Your data to be lost You allways should collect them by the output digitized points instruction OD. The OD command should follow any Digitize Command at once.

The Digitize Two Points with Box Cursor Instruction, DB

Description Another instruction for interactive use of the Advanced Pac Screen graphics device has been added to the command set with the digitize two points with box cursor instruction. Now it is possible to specify two points (lower-left and upper-right corners of a window) to select an area for any purpose You may intend.

Syntax DB;

Explanation On reception of the digitize two points instruction Advanced Pac Screen will show a cross-hair cursor to place the first reference point onto the graphics display when the mouse button is pushed and held down. The displayed cursor changes from cross-hair to box and can be moved around until the button is released. The execution of the DB command is finished and the data will be available by the output digitized points command, OD. The triggered positions are stored within the Advanced Pac Screen 2.0 until another digitize instruction has been recognized. It is not possible to digitize more than two points with the DB instruction . If you don´t want Your data to be lost You allways should collect them by two Output Digitized instructions, OD. The OD command should follow any Digitize Command at once.

The Digitize Point Instruction, DP

Description Another instruction for interactive use of the Advanced Pac Screen graphics device has been added to the command set with the digitize point instruction. Now it is possible to specify a point on the Graphic Screen with the mouse for any purpose You may intend. Together with the two other Digitize instructions DB and DL writing interactive plot programs is quite easy.

Syntax DP;

Explanation On reception of the digitize point instruction Advanced Pac Screen will show a cross-hair cursor which can be moved around until the button is pressed. The middle of the cross hair cursor is the point you specify on the graphic screen. The execution of the DP command is finished and the data will be available by the output digitized point command, OD. The triggered position is stored within the Advanced Pac Screen 2.0 until another digitize instruction has been recognized. It is not possible to digitize more than one point with the DP instruction . If you don´t want your data to be lost you allways should collect them with an Output Digitized instruction, OD. The OD command should follow any Digitize Command at once.

note !
If you send a DP instruction while the graphic page is not visible you might not see the cross-hair cursor but the IL Loop will be blocked until you pressed the mouse button.

The Output Digitized Points Instruction, OD

Description The Output Digitized Points Instruction, OD, is used to transfer the data produced by any Digitze command. There is an output buffer within PACPLOT, which is cleared by each Output Command as OP, OE, OW, ...

Therefore you must execute the OD command immediately after each Digitize command to make sure the data is not overwritten by another command. By executing OD the data will be put on the Loop and you must collect it with an appropriate input command. The coordinates are sent as ASCII numbers, parted by commas. At the end of the string the terminator is send.

```
10 !        This is a subroutine to transfer the contents of a data file
20 !        to the pacplot device of APS after a LI command.
30 !        This subroutine should be called immediately after the LI command
40 !        especially without any CR or LF characters between the LI and this
50 !        subroutine.
60 !
70 !
80 SUB LOGOIN
90 !
100 !
110 !       Declaration of the string; Blocksize is 256 Bytes
120 !
130 DIM A$[256]
140 !
150 !
160 !       Between the different blocks no CR or LF is allowed.
170 !
180 ENDLINE ''
190 !
200 !
210 !       PACPLOT device has to be addressed as printer
220 !
230 PRINTER IS :PACPLOT
240 !
250 !
260 !       Read the name of the data file
270 !
280 INPUT "FILENAME: ";N$
290 !
300 !
310 !       Open the file for reading
320 !
330 ASSIGN #1 TO N$
340 !
350 !
360 !       Reaching the end of the file will raise error 54.
370 !       That is the exit condition for the reading loop.
380 !       in this case the programm will continue in the line 'OFF ERROR'
390 !       to switch off the exception handling. Any error that occurs behind
400 !       that line will raise usual error messages again.
410 !
420 ON ERROR GOTO 540
430 !
440 !
450 !       Read one block (256 Bytes) in A$ , send it to PACPLOT and idle.
460 !
470 READ #1;A$
480 PRINT A$
490 GOTO 470
500 !
510 !
520 !      Here the program will continue after the 'End of File  error #54.
530 !
540 OFF ERROR
550 !
560 !
570 !      Switch ENDLINE sequence back to CR LF and send CR LF.
580 !
590 ENDLINE
600 PRINT
610 END SUB
```

```
10 !          This is a subroutine to transfer the output of PACPLOT after
20 !          a LO command to a data file that can be read later using the
30 !          LI instruction.The data file will be created in this subroutine.
40 !          No CR and LF or any other characters are allowed between the
50 !          LO instruction and calling this subroutine.
60 !
70 SUB LOGOOUT
80 !
90 !
100 !          Declaration of the buffer A$; Blocksize is 256Bytes
110 !
120 DIM A$[256]
130 !
140 !
150 !          Between the blocks no CR or LF is allowed.
160 !
170 ENDLINE ''
180 !
190 !
200 !          Flag 23 must be cleared to enable formatting
210 !
220 CFLAG -23
230 !
240 !
250 !          Read the name of the data file to be created.
260 !
270 INPUT "FILENAME: ";N$
280 !
290 !
300 !          Read the number of blocks to be transmitted from the HP-IL.
310 !
320 ENTER :PACPLOT USING "#,B";C
330 !
340 !
350 !          Create the data file and open it.
360 !
370 CREATE DATA N$,C,256
380 ASSIGN #1 TO N$
390 !
400 !
410 !          This is the transmitting loop. It reads data of PACPLOT
420 !          device from the HP-IL and puts it into the data file.
430 !
440 FOR X=1 TO C-1
450 ENTER :PACPLOT USING '#,256A';A$
460 PRINT #1;A$
470 NEXT X
480 !
490 !
500 !          The last block must be read without formatting because it
510 !          can be shorter than 256 Bytes. It is added to the data file.
520 !
530 ENTER :PACPLOT ;A$
540 PRINT #1;A$
550 !
560 !
570 !          Switch back the ENDLINE sequence to CR LF to enable normal
580 !          printer output.
590 !
600 ENDLINE
610 PRINT
620 END SUB
```

```
10 !         This is a subroutine to print graphics with a Thinkjet.
20 !         After a HT or LO command APS sends the contents of the selected
30 !         screen area on HP-IL. The first Byte contains the number of
40 !         256 Bytes blocks to be transmitted. In the following blocks all
50 !         controlling sequences that are needed for the Thinkjet are
60 !         enclosed. It is easy to save a header of a letter or a firm's
70 !         symbol for later use. You only have to save the data in a file in
80 !         your computer or on mass storage and send it to the Thinkjet later.
90 !         How to do this see the subroutine LOGOOUT and LOGOIN. They work this
100 !        way. The LOGOOUT subroutine can be called instead of this routine
110 !        to save the screen contents in a data file. For the other direction
120 !        from data file to Thinkjet the LOGOIN subroutine must be changed
130 !        a little.Not PACPLOT but the Thinkjet must be addressed as printer.
140 !
150 SUB PLOT
160 !
170 !        Declaration of the string; Blocksize is 256 Bytes
180 !
190 DIM A$[256]
200 !
210 !        Between the blocks no CR or LF is allowed.
220 !
230 ENDLINE ''
240 !
250 !        Read the number of blocks to be transmitted.
260 !
270 ENTER :PACPLOT USING "#,B";N
280 !
290 !        Thinkjet has to be addressed as printer.
300 !
310 PRINTER IS :HP2225B
320 !
330 !        Flag 23 must be cleared to enable formatting.
340 !
350 CFLAG -23
360 !
370 !        This is the transmitting loop. It reads data of PACPLOT'device
380 !        from HP-IL and transfers it blockwise to the Thinkjet. N-1 blocks
390 !        will be transmitted this way. The last block must be read without
400 !        formatting because it may be shorter than 256 Bytes.
410 !
420 FOR X=1 TO N-1
430 ENTER :PACPLOT USING "#,256A";A$ @ PRINT A$;
440 NEXT X
450 ENTER :PACPLOT ;A$ @ PRINT A$
460 !
470 !        Switch ENDLINE sequence back to CR LF.
480 !
490 ENDLINE
500 PRINT
510 PRINTER IS :PACPLOT
520 END SUB
```

The CENTRONICS Port of the Advanced Pac Screen 2.0

The standard interface on the multiport connector of your Advanced Pac Screen 2.0 is Centronics.

To connect a Centronics printer to the APS 2.0 you need a Centronics cable. you can order it from your local dealer or assemble it by yourself. The pinning of the Multiport where the cable will be plugged in is printed on the backside of your Advanced Pac Screen 2.0 . From the IL Loop the Centronics Interface is looked at as an extra device with following specifications:

DEVICE ID	-	PACCEN
ACCESSORY ID	-	46

According to your calculator there are several possibilities to adress the Centronics Port.

Using the **HP 71** as controller:

The first thing to do after switching on the HP 71 is executing a **RESET HPIL @ RESTORE IO** Instruction. Thereafter exist 3 ways to address the Centronics port.

```
PRINTER IS :PACCEN
PRINTER IS :%46
PRINTER IS :3  ( only if APS is the first unit in the loop)
```

All strings sent to the loop using the PRINT statement will be sent on to the printer via the CENTRONICS Port.

Using the **HP 41** as controller:

3	**ENTER**	the '3' is appropriate only if the
XEQ	**SELECT**	Advanced Pac Screen 2.0 is the 1.unit
		in the IL Loop.

All strings sent to the loop using the OUTA statement will be sent on to the printer via the CENTRONICS Port.

Using the **HP 75** as controller:
The first thing to do after switching on the HP 75 is executing a **RESTORE IO** command. Therafter you should assign names to the devices connected to the IL Loop by executing **ASSIGN IO** . Now the HP 75 will inform you how many Devices are connected to the Loop and asks you about the names you want to assign to them. If the Advanced Pac Screen 2.0 is the first unit in the Loop you might name the third device "**PR**" for printer. Now you can adress the CENTRONICS Port by **PRINTER IS ":PR"**. All strings sent to the loop using the PRINT statement will be sent on to the printer via the CENTRONICS Port.

Response to IL commands

Device ID	-	PACCEN
Accessory ID	-	46
Status ID	-	0 normal condition 163 Timeout, no printer
Clear	-	Timeout is cleared Printer Buffer is cleared Standard ESC Sequences (EPSON)
Trigger	-	The 19 Bytes following the Trigger CMD are interpreted as new ESC Sequence

Handshake - How a Centronics printer obtains data

Data are sent to the printer using a Busy Handshake. That means whenever the Printer cannot receive data it sets the Busy line high.

BUSY-	0V	printer ready to receive data
	5V	printer busy, or no printer connected

All data, which are sent to PACCEN will be transfered by the Advanced Pac Screen 2.0 to the printer likewise. In case data is sent to PACCEN by mistake, without any printer connected, the IL LOOP would be blocked. The only way to reactivate the loop would be to switch off/on APS 2.0.

Therefore a printer timeout is installed within PACCEN. If the printer interface signals BUSY longer than 30 seconds, it is assumed that an error ocurred and all data designated to PACCEN is not sent further on. So the data is lost but the loop handshake can go on. To put PACCEN back in the startup condition, you have to perform a DEVICE CLEAR on PACCEN.

Adjusting Advanced Pac Screen 2.0 to your printer

All new matrix printers support the printing of Dot Matrix Graphic. How this works in detail shall be explained here.

In the printing head there are 8 or more needles, who are pressed to the ribbon tape in front of it. Every time this happens a dot is drawn on the paper. Usually the printer receives one byte and undertakes all actions, necessary to print a character. But there also is the possibility to control the needles by yourself. This is necessary for printing characters which are not included in the printer`s character set, e.g. the HP custom characters. Previously the printer has to be told that this modus is wanted. That happens with so called ESC-Sequences. The following data are not printed as characters any more, but is used to controll the 8 needles. If you send one byte i.e. 5 = 1+4 after switching on the graphic`s modus, the lowest (1.) and (3.) needle from below are pressed to the ribbon tape ($5 = 2^0 + 2^2$ so the needles representing Bit 0 and Bit 2 are activated). Thereby two points are printed. With some pinters the needles from the top, but not the ones from below, would have been activated. In the following this difference will be referred to as the **most significant needles below** or **on top** .As this feature is different with several printers, it is adjustable within your Advanced Pac Screen 2.0

Most significant needle on top	**Most significant needle below**
O - 128	● - 1
O - 64	O - 2
O - 32	● - 4
O - 16	O - 8
O - 8	O - 16
● - 4	O - 32
O - 2	O - 64
● - 1	O - 128

Normal and Double Density while printing Graphic

With several printers there is the possibility to print graphic with double density. This is important if you wish to print the whole graphic screen image from APS 2 , because the paper width using single density sometimes is not sufficient. While printing HP custom characters, the normal density is used. In case your printer has just one graphic modus, you have to put in the ESC-sequence to switch it on twice. Once for graphic with single density and for graphic with double density in the ESC-sequence word.

Number of Graphic Bytes

Besides, the printer needs the number of bytes, which shall be interpreted as graphic data, since the printing has to continue normally after the custom characters or graphics have been printed. Even here different formats are necessary, which are selectable by the modus byte.

Most printers need the number of graphic bytes in hexadecimal format. That means:if e.g. 700 graphic bytes shall be printed, this number is divided into two parts and will be sent to the printer as shown.

700/256 = 2 fraction 188

1. Byte 188 , fraction of division
2. Byte 2 , 256 bytes

NEC printers and some Apple printers have to be supplied by ASCII characters for the number. If 700 bytes shall be sent, as shown above, the 4 bytes would be sent as follows:

1.-4. byte \`0700\`

Line-Feed

At the end of every line, which is printed, a CRLF (Carriage Returne/Line Feed) is sent to the printer, whereupon the printing head goes back to the beginning of a line and the paper is moved one line up.

Usually there is a space between the lines. This is unwanted while printing graphics. Therefore the linefeed interval has to be adjusted in a way, that there is no space between the lines. After the graphic is printed the original linefeed has to be selected to have an interval between the lines of the text. Space between lines of most printers can be adjusted by sending an ESC-sequence in steps of 1/72 or 1/144 inch ;8/72 inch is useful for graphic printing and 10/72 for text printing.

Before the graphic is printed, PACCEN sends to the printer the ESC- sequence for graphic linefeed; after having finished printing graphics, PACCEN sends the ESC-sequence for normal line spacing.

Printing Graphics and HP Custom Characters

PACCEN is able to print the HP custom characters. All characters with a Character Code between 1 and 32 and with character code 255 are not sent to the printer directly but are sent to the printer as Bit Image Graphic, except of not printable control codes as ESC, CR, LF,....

That means that the whole HP character set can be printed using a normal centronics printer. For that reason the printer has to be initialized by an ESC-sequence (an ESC-sequence is a series of control codes), which signalizes that e.g. the following 6 bytes shall be interpreted as graphic, as it is necessary for printing the characters. The ESC-sequences which are necessary for Epson FX or RX 80 or compatible printers are already included in APS 2.0 as default.

It will be explained in the following, how the APS 2.0 can be adjusted to different printers. In addition a **Transparent Mode** can be selected where all data is passed to the printer unchanged. If you want to send characters with an ASCII code from 0 to 32 or 255 you should select the **Transparent Mode.**

How to build the ESC Command word for PACCEN

From your printer`s handbook you choose following ESC-sequences:

	Function	Default	Number of Bytes
1.	most significent needle	on top	1
	HP custom characters	printed, no transparent mode	
	Nr of Graphic Bytes	Hex Format	
2.	line spacing graphic	ESC A; CHR$(8)	5
3.	graphic double density	ESC L	4
4.	graphic single density	ESC K	4
5.	line space text	ESC A; CHR$(10)	5

Structure of APS 2.0 Control Codes

The first byte of control codes defines the mode of the printer. With this byte you can select different operating modes explained above.

HP Custom Characters	=	0	or
Transparent Mode	=	1	and
Most Significant Needle Above	=	0	or
Most Significant Needle Below	=	2	
Number of Graphic Bytes ASCII	=	0	or
Number of Graphic Bytes HEX	=	4	

The total of your choice defines the mode byte. If you wish for instance:

Print HP custom characters	0
most significant needle below	+2
number of graphic bytes in HEX format	+4
mode byte	=6

4 or 5 bytes are provided for each of the 4 ESC-sequences, the last byte always has to be zero. In case you need less, you have to fill the spaces between the ESC sequences with zero bytes (CHR $ (0)).

Sending the Control Sequence to PACCEN

By the HPIL command **TRIGGER PACCEN** it is signaled to PACCEN, that the control sequences will follow. The 19 bytes following are interpreted as the new control sequences. As an example, a short program is supplied, which sends the standard control sequences to PACCEN.

```
PRINTER IS :PACCEN                          ! PACCEN is adressed as printer

TRIGGER  PACCEN                             ! the  next 13  Bytes are
                                            ! interpreted as control word

PRINT CHR$(0);                              ! Mode Byte

PRINT CHR$(27);"A";CHR$(8);CHR$(0);CHR$(0)
                                            ! Line spacing  8/72 Zoll for
                                            ! printing graphics

PRINT CHR$(27);"L";CHR$(0);CHR$(0);         ! Graphic double density

PRINT CHR$(27);"K";CHR$(0);CHR$(0);         ! Graphic single density

PRINT CHR$(27);"A";CHR$(10);CHR$(0);CHR$(0)
                                            ! Line spacing  10/72 Zoll for
                                            ! printing text
```

Additions to the Programmer's Manual for the Advanced Pac Screen V.2.0

Page 13 Two additional commands:-

ESC r -The characters with a code number larger than 127 are interpreted as Roman8. They are no longer printed inverse (default).

ESC h -The characters with a code number larger than 127 are printed inverse.

BELEGUNG MOUSEPORT:

Pin-Nr.:	1	Masse
	2	+5V
	3	frei
	4	Horizontal 1
	5	Horizontal 2
	6	frei
	7	Taster
	8	Vertikal 1
	9	Vertikal 2