

CHAPTER 4

HP-GL/2 GRAPHICS LANGUAGE

CONTENTS

1. COMMAND LIST	4
2. INTRODUCTION	6
3. TERMINOLOGY	7
3.1. Picture Frame.....	7
3.2. Anchor Point.....	7
3.3. Graphics Window.....	7
3.4. Hard Clip Limits.....	7
3.5. Soft Clip Limits.....	7
3.6. Graphics Units.....	7
3.7. User Units.....	7
3.8. Current Units.....	7
3.9. Pens.....	7
3.10. Absolute and Relative Plotting.....	8
3.11. Fills.....	8
4. COMMAND SYNTAX	9
4.1. Mnemonic.....	9
4.2. Parameters.....	9
4.3. Separators.....	9
4.4. Terminator.....	9
5. THE HP-GL GRAPHICS WINDOW	10
5.1. Units.....	10
5.2. The Scaling Points P1 and P2.....	10
5.3. Scaling.....	10
5.4. Transformation.....	10
6. PREPARING TO PRINT GRAPHIC IMAGES	11
6.1. Setting Up the Graphics Window.....	11
6.2. Using Existing Image Plots.....	11
6.3. Printing an Image.....	11
6.4. Entering HP-GL/2 Mode.....	11
6.5. Quitting HP-GL/2 Mode.....	11
7. COMMANDS	12
7.1. Initialize Commands.....	12
7.2. Plot Area and Unit Setting Instructions.....	14
7.3. Pen Control and Plot Instructions.....	18
7.4. The polygon group.....	25
7.5. Plot Function Instructions.....	30

7.6. Character Plot Instructions..... 38
7.6.1. Introduction39
8. INDEX..... 52

1. COMMAND LIST

DF	Default set instruction	12
IN	Initialize set instruction	13
IP	Input scaling point	14
IR	Input relative scaling points	14
SC	Scale	15
IW	Window	17
RO	Rotate coordinate system	17
PU	Pen up	18
PD	Pen down	18
PA	Plot absolute	19
PR	Relative coordinate pen move	19
AA	Draw absolute arc	20
AR	Draw relative arc	21
AT	Absolute arc three point	21
RT	Relative arc three point	22
PE	Polyline encoded	22
BR	Bezier relative	24
BZ	Bezier absolute	24
PM	Polygon mode	25
EA	Edge rectangle absolute	25
ER	Edge rectangle relative	26
EP	Edge polygon	26
EW	Edge wedge	26
RA	Fill rectangle absolute	27
RR	Fill rectangle relative	28
WG	Fill wedge	28
FP	Fill polygon	29
AC	Anchor corner	30
FT	Fill type	31
LA	Line attribute	32
LT	Line type selection	33
PW	Pen width	34
RF	Raster fill definition	34
SM	Symbol mode	35
SP	Select pen	35
SV	Screened vectors	35
TR	Transparency mode	36
UL	User-defined line type	37
WU	Select unit for pen width	37
SD	Define standard font	39
AD	Define alternate font	42
SS	Select standard font	42
SA	Select alternate font	42
FI	Select primary font	43
FN	Select secondary font	43
LB	Character plot	43
DT	Define label terminator	44
LO	Label origin	44

DI	Absolute direction	45
DR	Relative direction	46
DV	Define variable text path	47
CP	Relative character movement	47
CF	Character fill mode	48
SI	Set absolute character size	48
SR	Set relative character size	49
SL	Character slant	49
SB	Scalable or bitmap fonts	50
ES	Extra space	50
TD	Transparent data	50

2. INTRODUCTION

Some of the HL series printers can be used in HP-GL/2 graphics mode to produce vector graphics output using the commands of the HP-GL/2 graphics language or to print out existing HP-GL/2 format images.

HP-GL/2 is a part of the PCL emulation, which includes graphics command.

The Default plotting measurement unit in the HP-GL/2 graphics mode is 1/1016" (0.025 mm).

3. TERMINOLOGY

3.1. Picture Frame

The printer frame is the rectangular area of the page on which graphic output can be printed. This is based on the logical page defined for the selected paper size.

3.2. Anchor Point

The anchor point of the picture frame is the top left hand corner of the frame. This can be moved on the page using PCL language page control commands.

3.3. Graphics Window

The graphics window is the rectangular area of the page on which you can generate graphic output using HP-GL/2 commands. Initially the graphics window is the same as the picture frame but may be altered using the HP-GL/2 IW command.

3.4. Hard Clip Limits

Hard clip limits are the physical restrictions which define where printed output can appear on a page.

3.5. Soft Clip Limits

Soft clip limits are the user-defined restrictions which define where printed output can appear on a page.

3.6. Graphics Units

The initial coordinate system employs units of 1/1016" in both the x- and y-axes.

3.7. User Units

The HP-GL/2 graphics language uses a coordinate system. You can impose your own x- and y-scale units on this system using the SC command.

3.8. Current Units

The units in use at a given time are called the current units. Current units are either user units or graphics units, depending on whether scaling has been turned on or not. The SC command is used to turn scaling on, that is, to make user units the current units.

3.9. Pens

By convention, certain HP-GL/2 graphics language commands are thought of as using a pen, which can be either up or down. To produce plotted output the pen must be down. To move to a location on the page without producing output the pen must be in the up position. Some HP-GL/2 commands require you to specify the pen position before you use them, others have implied pen positions built into them, so that you do not need to specify "Pen up" or "Pen down".

3.10. Absolute and Relative Plotting

Drawing commands are of two distinct types: absolute commands, in which the coordinates specified are the absolute user or graphics coordinates and relative commands, whose coordinates are relative to the coordinate position at which the previous graphics command terminated.

3.11. Fills

You can fill in the shapes you have drawn with the plotting commands by using the fill commands. A number of different shading options are available.

4. COMMAND SYNTAX

An HP-GL and HP-GL/2 command can consist of up to four items: a mnemonic, a parameter field, a separator and a terminator.

4.1. Mnemonic

The two-letter mnemonic is the name of the command and should help remind you of its function.

4.2. Parameters

Some commands have parameters following the instruction mnemonic. Multiple parameters must have separators between them. Parameters are shown in italics. Optional parameters are surrounded by square brackets. Parameters that can be repeated are followed by "...".

- Integer - integers are allowed in the range -2^{30} and $2^{30} - 1$. Real numbers specified for a parameter that should be an integer are rounded to the nearest integer.
- Clamped integer - integers are allowed in the range -32768 and 32767. Parameters outside this range are converted (clamped) to the nearest integer within the range, for example 40,000 will be clamped to 32767. Non-integers are converted to the nearest integer.
- Real number - real numbers whose integer part is in the range -2^{30} to $2^{30} - 1$. Accuracy to at least 6 significant digits is guaranteed. If a number outside the range is specified the entire command is ignored. The decimal point can be omitted when the number has no fractional part.
- Clamped real number - real numbers whose integer part is in the range -32768 to 32767. Parameters outside this range are converted (clamped) to the nearest real number within the range. The decimal point can be omitted when the number has no fractional part.

Text parameters are referred to as labels and can consist of any sequence of characters.

4.3. Separators

If an HP-GL/2 instruction accepts more than one parameter, you must put spaces or a comma between the parameters to separate them.

4.4. Terminator

Most commands are terminated implicitly by the first letter of the next instruction mnemonic. A semi-colon may be used to explicitly terminate a command. However, the final command issued before quitting HP-GL/2 mode must be terminated with a semi-colon. Since a terminator is almost always optional it is shown surrounded by square brackets.

5. THE HP-GL GRAPHICS WINDOW

5.1. Units

When you first enter graphics mode, the prevailing coordinate system has its origin in the bottom left hand corner of the graphics window and has x- and y-units of 1/1016". Using the SC command you can divide the axes into more convenient units. The size of the graphics window remains unaltered.

5.2. The Scaling Points P1 and P2

Many instructions use two reference scaling points, P1 and P2. Initially these are in the bottom left and top right hand corners of the graphics window respectively. You can alter the position of these points relative to the graphics window, or relative to one another, by using the IP or IR commands.

5.3. Scaling

You can easily scale images up or down by increasing or decreasing the distance between P1 and P2. You can do this both with existing images that are to be imported and with images that are generated by a string of HP-GL and HP-GL/2 commands, identical commands can be used to create the same image in different sizes.

5.4. Transformation

You can transform images horizontally or vertically by changing the relative positions of P1 and P2. You can do this both with existing images that are to be imported and with images that are generated by a string of HP-GL/2 commands.

6. PREPARING TO PRINT GRAPHIC IMAGES

6.1. Setting Up the Graphics Window

Before entering HP-GL/2 mode you will probably want to set the dimensions and position of the PCL picture frame and hence the graphics window. (Remember that when you enter HP-GL/2 mode they are initially the same). To do this you will need to send some preparatory PCL commands while in LaserJet mode. Use the Esc*c0T command to position the picture frame and the Esc*c#X and Esc*c#Y commands to determine its horizontal and vertical dimensions respectively. These commands are described fully in the sub-section Vector Graphics in the LaserJet section of this manual. (Section 2)

If you do not specify a position and dimensions for the picture frame, the default picture frame is used. This is described in the sub-section 'The Page in LaserJet mode' section of this manual.

6.2. Using Existing Image Plots

If you are importing an existing HP-GL/2 plot you can easily determine the size and position on the page of the resulting printed graphic. Images are automatically scaled and fitted to the picture frame (the HP-GL/2 graphics window) in either of the following two cases.

- 1) If you use the Esc*c#K and Esc*c#L PCL commands to specify the width and height of the original plot before entering HP-GL/2 mode. These commands are explained in the sub-section 'Vector Graphics in LaserJet mode' section of this manual. Only use these commands if you are importing an existing plot.
- 2) If the imported image code is page-size independent.

An image is page-size independent if its HP-GL/2 code satisfies the following conditions.

- User units are used exclusively, that is, an SC command is used before any drawing commands are issued. Note in particular that if the graphics window is changed from the default using the IW command, it must be specified in user units.
- No absolute units are used or implied, that is, drawing commands use relative movement only and character size, pattern length and pen width are always specified as relative measurements. Commands like IP and PA, which imply the use of absolute units even if none are explicitly supplied, should not be used at all.

If you import an image without fulfilling these two criteria, that is, the image is not page size independent and you do not specify its horizontal and vertical plot sizes, the plot will be drawn actual size. Any part of the plot that falls outside the effective window will not appear on the printed page.

6.3. Printing an Image

When you have carried out any necessary preliminary steps that your particular application requires, as described in the preceding sections, enter HP-GL/2 mode using the Esc%#B command and send the sequence of HP-GL/2 commands to generate the image.

6.4. Entering HP-GL/2 Mode

To enter graphics mode from PCL printer control language use the Esc%#B command. Set # to 0 if you want the pen to start from the previous HP-GL/2 pen position. Set # to 1 if you want to the pen to start from the current PCL cursor position. You must specify either one setting or the other.

6.5. Quitting HP-GL/2 Mode

To quit graphics mode use the Esc%#A command. Set # to 0 if you want to place the cursor at the previous PCL cursor position. This is the default setting. Set # to 1 if you want to place the cursor at the current HP-GL/2 pen position.

7. COMMANDS

7.1. Initialize Commands

DF - Default set instruction

DF [;]

- Returns the graphics mode to default conditions. The following are the default settings.

Function	Equivalent Instruction	Condition
Plot mode	PA;	Absolute plotting
Absolute direction	DI 1,0;	Horizontal
Line type	LT;	Solid line
Line pattern length	LT;	4% of distance from P1 to P2
Anchor corner	AC;	Lower left corner of picture frame
Line Attributes	LA;	Butt cap, mitered joints, miter limit = 5
User-Defined line type	UL;	All line types set to default
Window	IW;	Set at limits of printable area according to paper size
Symbol mode	SM;	Off
Character Fill Mode	CF ;	Solid, Fill, no edging
Character set	SS;	Standard character set selected
Scalable or Bitmap Fonts	SB 0;	Scalable font only
Character slant	SL 0;	0 degrees
Scaling	SC ;	Off
Label terminator	DT chr\$(3)	chr\$(3), non-printing
Raster fill	RF;	Solid black
Fill type	FT;	solid fill
Transparency mode	TR1;	Transparency mode on
Screened vectors	SV;	No screening
Polygon mode	PM0PM2;	Polygon buffer empty
Standard font	SD;	Stick font
Alternate font	AD;	Stick font
Character size transformation	SI;	Off
Label origin	LO1;	Current pen location
Text path	DV;	Left to right with normal line feed
Extra space	ES;	No extra space
Transparent data	TD;	Normal printing mode

NOTE: The location of the scaling points(P1, P2) remains constant.

IN - Initialize set instruction**IN[;]**

- Returns the graphics mode to initial conditions;

Function	Equivalent Instruction	Condition
Plot mode	PA;	Absolute plotting
Relative character direction	DR 1,0;	Horizontal
Line type	LT;	Solid line
Line pattern length	LT;	4% of distance from P1 to P2
Window	IW;	Set at limits of printable area according to paper size
Symbol mode	SM;	Off
Anchor corner	AC;	Lower left corner of picture frame
Line attributes	LA;	Butt caps, mitered joins, miter limit = 5
User-defined line type	UL;	All line types set to default
Pen	SP;	Select white pen
Pen position	PA0,0;	HP-GL/2 origin
Character set	SS;	Standard character set selected
Character size transformation	SI;	Off
Scalable or bitmap fonts	SB0;	Scalable only
Character slant	SL 0;	0 degrees
Character fill	CF;	Solid fill
Scaling	SC ;	Off
Label terminator	DT chr\$(3)	chr\$(3)
Raster fill	RF;	Solid black
Fill type	FT;	Bi-directional fill, type 1
Transparency mode	TR1;	Transparency mode on
Screened vectors	SV;	No screening
Polygon mode	PM0PM2;	Polygon buffer empty
Standard font	SD;	Stick font
Alternate font	AD;	Stick font
Pen condition	PU;	Pen up
Pen width type	WU;	Metric
Pen width	PW;	Set at 0.35 mm
Label origin	LO1;	Current pen location
Text path	DV;	Left to right with normal line feed
Rotation	RO;	Set at 0 degrees
Extra space	ES;	No extra space
Transparent data	TD;	Normal printing mode
Scaling points	IP;	Initialized according to paper size

7.2. Plot Area and Unit Setting Instructions

Instruction	Function
IP	Scaling point
IR	Input relative scaling points
SC	Scale
IW	Input window
RO	Rotate coordinate system

IP - Input scaling point

IP [P1_x, P1_y [, P2_x, P2_y]] [;]

P1_x ; x coordinate of P1 P1_y ; y coordinate of P1

P2_x ; x coordinate of P2 P2_y ; y coordinate of P2

- The coordinates used are absolute values in graphics units.
- Set the location of the scaling points(P1,P2).
- Coordinate values for P1_x,P1_y,P2_x and P2_y are given as integer numbers.
- You can set P1 and P2 to be anywhere that the range of allowable coordinates permits. However, only parts of resulting images that lie within the effective window will be printed.
- The default location of P1 is the lower left corner and P2 is the upper right corner of the PCL picture frame.
- Using this instruction without a parameter field initializes the scaling points(P1,P2) to the default location.
- P2_x and P2_y may be omitted. If P2_x and P2_y are omitted, P2 is set automatically so as not to alter the distance between P1 and P2.
- If P2_x and P2_y are omitted P2 is set to be in the same position relative to P1 that it was before the command was invoked.
- The IP command remains in effect until another IP command, an IR command or an IN command is used.

IR - Input relative scaling points

IR [P1_x, P1_y [, P2_x, P2_y]] [;]

P1_x ; x-coordinate of P1 P1_y ; y-coordinate of P1

P2_x ; x-coordinate of P2 P2_y ; y-coordinate of P2

- The coordinate values are percentages relative to the bottom left and top right hand corners of the picture frame.
- Coordinate values for P1_x, P1_y, P2_x, and P2_y are treated as clamped real numbers.
- Using this instruction without parameters sets the scaling points (P1, P2) to their initial positions, the bottom left and top right hand corners of the picture frame.
- If P2_x and P2_y are omitted P2 is set to be in the same position relative to P1 that it was before the command was invoked.
- You can set P1 and P2 to be anywhere that the range of allowable coordinates permits. However, only parts of resulting images that lie within the effective window will be printed.
- The IR command remains in effect until another IR command, and IP command or an IN command is used.

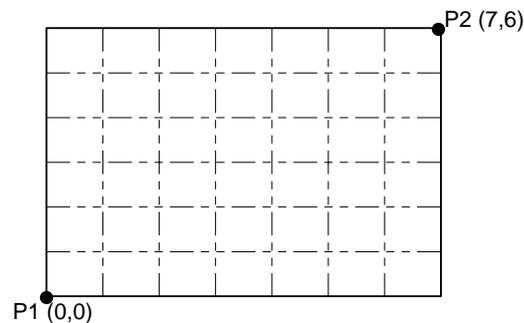
SC - Scale

<HP-GL/2>

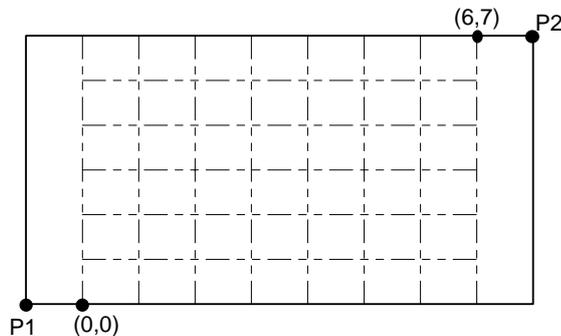
SC [X_{min} , X_{max} , Y_{min} , Y_{max} [, *type*]] [;] (*Type* = 0)SC [X_{min} , X_{max} , Y_{min} , Y_{max} , *type* [*left*, *bottom*]] [;] (*Type* = 1)SC [X_{min} , X_{factor} , Y_{min} , Y_{factor} [, *type*]] [;] (*Type* = 2) X_{min} ; X coordinate of P1 X_{max} ; X coordinate of P2 Y_{min} ; Y coordinate of P1 Y_{max} ; Y coordinate of P2*Type* ; type of scaling*Left* ; percentage of unused space to be left to the left of the scaling area*bottom* ; percentage of unused space to be left below the scaling area X_{factor} ; number of graphics units per user unit on the x-axis. Y_{factor} ; number of graphics units per user unit on the y-axis.

- This command turns scaling on and establishes a user unit coordinate system. The coordinate system is specified relative to the points P1 and P2. The origin need not necessarily be at P1. Three different types of scaling are available.
- The technical terms, user unit and graphics unit, as used in this manual are defined as follows:

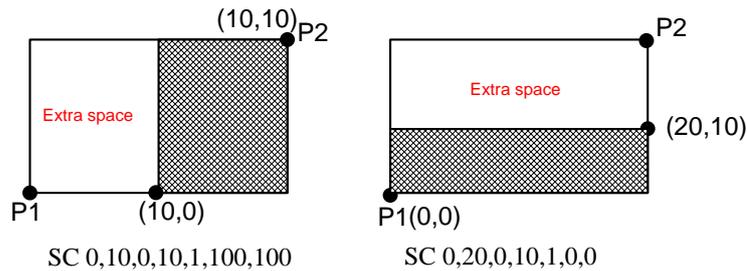
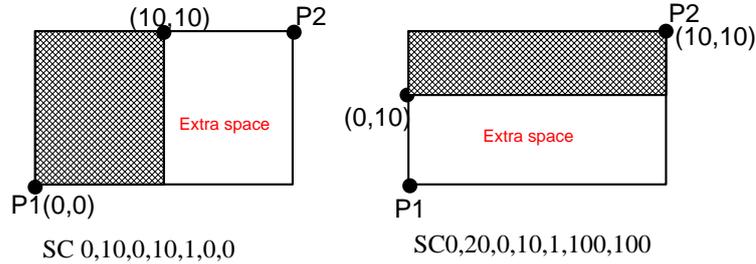
User unit	: the unit of the coordinates set by the SC instruction.
Graphics unit	: the unit(1/1016 of an inch) of the coordinates not set by the SC instruction.
- All coordinates are in user units.
- Coordinate values for X_{min} , X_{max} , Y_{min} , and Y_{max} are given as real numbers.
- X_{min} , Y_{min} , X_{max} and Y_{max} specify the range of the X- and Y-axes. X_{min} and Y_{min} value should be P1 and X_{max} and Y_{max} should be P2.
- X_{min} must be different from X_{max} , and Y_{min} must be different from Y_{max} .
- Using this instruction without a parameter field turns the scaling off.
- *Type* can be 0, 1 or 2, which select scaling types known as anisotropic, isotropic and point factor scaling respectively.
- *Type* 0 scaling (anisotropic scaling, the default) allows you to specify different sized x-axis and y-axis units. Hence the area specified by X_{min} , X_{max} , Y_{min} , and Y_{max} will occupy the entire area defined by P1 and P2.



- left and bottom settings have no effect in *type* 0 scaling.
- *Type* 1 scaling (isotropic scaling) forces x-axis and y-axis units to be the same size. As a result the area specified by X_{min} , X_{max} , Y_{min} , and Y_{max} (the isotropic area) may not fit exactly into the rectangular area defined by P1 and P2.



- With *type 1* scaling you can specify the percentage of unused space that is to lie below, or to the left of, the isotropic area.
- *left* specifies the percentage of unused space you want to lie to the left of the isotropic area and can be from 0 to 100. If you do not specify the value for *left*, it is automatically set to the default value = 50%.
- *bottom* specifies the percentage of unused space you want to lie below the isotropic area and can be from 0 to 100. If you do not specify the value for *bottom*, it is automatically set to the default value = 50%.



- You must either specify both *left* and *bottom*, or specify neither. However, the printer can only use one of these two parameters at a time, depending where the extra space is on a page. \
- If you do not specify a percentage for unused space to the *left* or at the *bottom*, the isotropic area is centred in the rectangle defined by P1 and P2.
- *Type 2* scaling (point factor scaling) specifies the ratio of graphics units to user units and sets user units coordinates for P1.
- X_{factor} specifies the number of graphics units that comprise a single user x-axis unit.
- Y_{factor} specifies the number of graphics units that comprise a single user y-axis unit.
- Valid values for X_{factor} and Y_{factor} are real numbers.
- An SC command remains in effect until another SC command is issued, until default settings are restored with the DF command or until the printer is initialized with an IN command.

```

10 '-Scale-'
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;"
60 '-- Anisotropic scaling --
70 LPRINT "PA1500,6000;EA3500,8000;"
80 LPRINT "IP1500,6000,3500,8000;SC0,200,0,100,0;PA50,50;CI50;"
90 LPRINT "IN;PA1500,3000;EA3500,5000;"
100 LPRINT "IP1500,3000,3500,5000;SC0,100,0,200,0;PA50,50;CI50;"
110 LPRINT"IN;PA1300,8200;DT#;LBAnisotropic scaling#";
120 '--Isotropic scaling--
130 'Left,Bottom= 0,0
140 LPRINT "PA4500,6000;EA6500,8000;"
150 LPRINT"IP4500,6000,6500,8000;SC0,200,0,100,1,0,0;PA50,50;CI50;"
160 'Left, Bottom = 100,100
170 LPRINT "IN;PA4500,3000;EA6500,5000;"
180 LPRINT"IP4500,3000,6500,5000;SC0,100,0,200,1,100,100;PA50,50;CI50;"
190 LPRINT "IN;PA4000,82;DT#;LBIotrop00ic scaling#";
200 LPRINT CHR$(27); "%0A";
210LPRINT CHR$(27); "E";
220 END

```

<Sample 21>

IW - Window**IW** [x_1 , y_1 , x_2 , y_2] [;] x_1 -Window lower left X coordinate y_1 -Window lower left Y coordinate x_2 -Window upper right X coordinate y_2 -Window upper right Y coordinate

- This instruction sets the window inside which drawing can be performed (the soft clip limits).
- Coordinates are specified in current units.
- Using this instruction with no parameters sets the window to be the same as the picture frame.
- You can set the corners of the window to lie outside the picture frame. However, only the portion of the image which lies within both the picture frame and the window will appear on the page.
- If scaling is turned on with the SC command before the window is defined with the IW command, you can move the window on the physical page by specifying different locations for P1 and P2 with the IP and IR commands. The user unit coordinates that bound the window remain the same.
- If an SC command is used after the IW window definition, the window is fixed on the physical page and does not move when subsequent IP and IR commands are invoked.
- An IW command remains in effect until another IW command is issued, until default settings are restored with the DF command or until the printer is initialized with an IN command.

```

10 '--Input Window--
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27);"E";
40 LPRINT CHR$(27);"%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "SI0.2,0.35,PA2000,3200;;LBTHIS IS AN EXAMPLE OF IW";CHR$(3)
70 LPRINT "IW3000,1300,4500,3700;PD2000,1700;LBTHIS IS AN EXAMPLE OF
IW";CHR$(3)
80 LPRINT "PU3000,1300;PD4500,1300,4500,3700;PD3000,3700,3000,1300;PU;"
90 LPRINT CHR$(27);"%0A";
100 LPRINT CHR$(27);"E";
110 END

```

<Sample 22>

RO - Rotate coordinate system**RO** [q] [;] q : Angle in degrees through which the coordinate system is rotated.

- This instruction is used to rotate the graphics coordinate system anti-clockwise.
- Using this instruction with no parameter sets the rotation of the coordinate system to 0 degrees (default value).
- The pen position does not change.
- P1 and P2 rotate with the coordinate system and retain the same coordinate values.
- To relocate P1 and P2 at the lower left and upper right hand corners of the picture frame, use the IP or IR commands.
- If you have previously used an IW command, the window is also rotated. Any part of the window that falls outside the hard clip limits will be clipped.
- The command remains in effect until another RO command is used or until the printer is initialized with the IN instruction.
- Valid values of q are 0, 90, 180 or 270.

```

10 '--Rotate Coordinate System--
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27);"E";
40 LPRINT CHR$(27);"%0B";
50 LPRINT "IN;SP1;";
60 FOR I = 0 TO 270 STEP 90
70 LPRINT "RO"; I; "PA1000,2000;PD3000,2000,2000,3000,1000,2000;PU;"
80 LPRINT "LB RO ="; I; CHR$(3)
90 LPRINT "PA600,1500;LBRotate Coordinate System";CHR$(3)
100 NEXT I
110 LPRINT CHR$(27);"%0A";
120 LPRINT CHR$(27);"E";
130 END

```

<Sample 23>

7.3. Pen Control and Plot Instructions

Instruction	Function
PU	Pen Up
PD	Pen Down
PA	Plot Absolute
PR	Relative Coordinate Pen Move
AA	Absolute Arc Plot
AR	Relative Arc Plot
AT	Absolute Three Point Arc
RT	Relative Three Point Arc
PE	Polyline Encoded
CI	Circle
BR	Bezier Relative
BZ	Bezier Absolute

PU - Pen up

PU [X, Y [, ...]] [;]

X ; X coordinate of the cursor movement destination

Y ; Y coordinate of the cursor movement destination

- X and Y are either relative or absolute, depending on whether a PA or a PR was the last plot command executed. The absolute coordinates are set as default.
- You can specify as many coordinate pairs as you wish. The cursor moves to each point in turn without drawing.
- Using this instruction without a parameter field raises the pen without changing the cursor position.
- Coordinates are specified in the currently selected unit.
- In symbol mode PU draws the specified symbol at each point in the parameter list.
- In polygon mode the specified coordinate pairs are placed in the polygon buffer.
- If an odd number of coordinates are specified, the final (unpaired) coordinate is ignored.

PD - Pen down

PD [X, Y [, ...]] [;]

X ; X coordinate of the cursor movement destination

Y ; Y coordinate of the cursor movement destination

- X and Y are either relative or absolute, depending on whether a PA or a PR was the last plot command executed. The absolute coordinates are the default setting.
- You can specify as many coordinate pairs as you wish. The cursor moves to each point in turn, drawing lines on the page.
- Using this instruction without a parameter lowers the pen without changing the cursor position.
- Coordinates are specified in the currently selected unit.
- In symbol mode PD draws the specified symbol at each defined point on the x- and y-axis.
- In polygon mode the specified coordinate pairs are placed in the polygon buffer.
- If an odd number of coordinates are specified, the final (unpaired) coordinate is ignored.

PA - Plot absolute**PA** [X, Y [, ...]] [;]

X ; X coordinate of the cursor movement destination

Y ; Y coordinate of the cursor movement destination

- Coordinates are absolute values in user or current units.
- Whether used with or without parameters the command establishes absolute plotting as the plotting mode.
- The command moves the cursor to the specified coordinates, drawing straight lines, only when the pen is down.
- In symbol mode PA draws the specified symbol at each point in the parameter list.
- In polygon mode the specified coordinate pairs are placed in the polygon buffer.
- If an odd number of coordinates are specified, the final (unpaired) coordinate is ignored.

```

10 ' -- Plot Absolute --
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA2000,6000;PDO,6000,2000,7500,2000,6000;PU2500,6000;";
70 LPRINT "PAPD4500,6000,2500,7500,2500,6000;PU10365,500;";
80 LPRINT CHR$(27); "%0A";
90 LPRINT CHR$(27); "E";
100 END

```

<Sample 24>

PR - Relative coordinate pen move**PR** [X, Y [, ...]] [;]

X ; X coordinate of the cursor movement destination

Y ; Y coordinate of the cursor movement destination

- Coordinates used are relative values in current units.
- Whether used with or without parameters the command establishes relative plotting as the plotting mode.
- The command moves the cursor to the specified coordinates, drawing straight lines, only when the pen is down.
- You can specify any number of coordinated pairs and plotting is performed in the order the coordinates are specified.
- In symbol mode PR draws the specified symbol at each point in the parameter list.
- In polygon mode the specified coordinate pairs are placed in the polygon buffer.
- If an odd number of coordinates are specified, the final (unpaired) coordinate is ignored.

```

10 ' - Plot Relative -
20 WIDTH "LPT1:", 255
20 LPRINT CHR$(27); "E";
30 LPRINT CHR$(27); "%0B";
40 LPRINT "IN;SP1;"
50 LPRINT "PA5000,4500,;PDPR-2000,0,2000,2000,0,-2000;PU500,0;"
60 LPRINT "PD2000,0,-2000,2000,0,-2000;PU;"
70 LPRINT CHR$(27); "%0A";
80 LPRINT CHR$(27); "E";
90 END

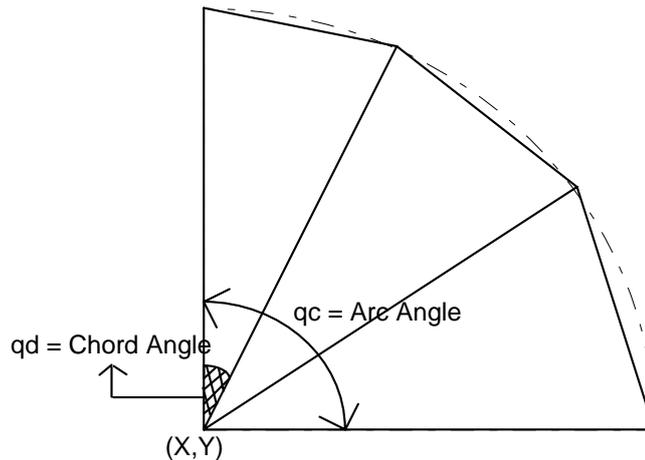
```

<Sample 25>

AA - Draw absolute arc**AA** [*x*, *y*, *qc* [, *qd*]] [;]

X ; Arc centre X coordinate *Y* ; Arc centre Y coordinate
qc ; Arc angle in degrees *qd* ; Chord angle in degrees

- Coordinates are absolute coordinates in current units.
- Starting from the current position, this command plots an arc centred on the absolute coordinates *X*, *Y* having the specified arc angle and chord angle, with the radius being the distance between the current position and the point *X*,*Y*.
- After execution of the command the cursor position is at the opposite end of the arc from the point at which plotting started.
- Plotting takes place only when the pen is down.
- If the command is invoked with the pen up, plotting is not performed, but the cursor position moves to what would have been the plot end point.



- The value for *qc* is a clamped real number.
- When *qc* is positive, plotting is counterclockwise.
- When *qc* is negative, plotting is clockwise.
- The value for *qd* is a clamped real number in the range 0.5° to 180°. The default value is 5°.

```

10 '-- Draw Absolute Arc --
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27);"E";
40 LPRINT CHR$(27);"%0B";
50 LPRINT "IN;SP1;IP1000,1000,6000,6000;";
60 LPRINT "SC0,100,0,100;";
70 LPRINT "PA0,30;";
80 LPRINT "PD;PA0,45;AA0,50,180;PA0,70;";
90 LPRINT "AA0,100,900;PA100,55;AA100,50,180;PA100,30;";
100 LPRINT "AA100,100,90;PA45,100;AA50,100,180;PA80,100;";
110 LPRINT "AA100,0,90;PA55,0;AA50,0,180;PA30,0;AA0,0,90;";
120 LPRINT "PU;PA50,50,CI20;";
130 LPRINT CHR$(27); "%0A";
140 LPRINT CHR$(27); "E";
150 END

```

<Sample 26>

AR - Draw relative arc**AR X, Y, qc(, qd) [;]**

X ; Arc centre X coordinate Y ; Arc centre Y coordinate
 qc ; Arc angle in degrees qd ; Chord angle in degrees

- Coordinates are relative values in current units.
- Starting from the current cursor position the command plots an arc whose centre is at the relative coordinate position (X,Y) and which has the specified arc and chord angles. The radius of the arc is the distance between the current position and the point (X,Y).
- After execution of the command the cursor position is at the opposite end of the arc from the point at which plotting started.
- Plotting takes place only when the pen is down.
- If the command is invoked with the pen up, plotting is not performed, but the cursor position moves to what would have been the plot end point.
- The value for qc is a clamped real number.
- When qc is positive, plotting is counterclockwise.
- When qc is negative, plotting is clockwise.
- The value for qd is a clamped real number in the range 0.5° to 180°. The default value is 5°.

```

10 '--Draw Relative Arc --
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;IP1000,1000,6000,6000;";
60 LPRINT "SC-100,100,-100,100;";
70 LPRINT
   "PA-100,70;PD;PR30,0;AR0,-70,-90,15;AR70,0,90;PR60,0;PU;";
80 LPRINT CHR$(27); "%0A";
90 LPRINT CHR$(27); "E";
100 END

```

<Sample 27>

AT - Absolute arc three point**AT X₁, Y₁, X₂, Y₂ [, qd] [;]**

X₁ ; x-coordinate of intermediate point Y₂ ; y-coordinate of intermediate point
 X₂ ; x-coordinate of end point y₂ ; y-coordinate of end point
 qd ; the chord angle used to draw the arc

- All coordinates are absolute and in current units.
- The command draws an arc from the current position, through the specified intermediate position and finishing at the specified end position.
- After completion of the command the current cursor position is at the end of the arc.
- An arc can be drawn clockwise or anti-clockwise.
- If the current scaling mode is not isotropic, the arc may be elliptical rather than circular.
- The chord angle must be in the range 0.5° to 180°.
- The default chord angle is 5°.

```

10 '--Absolute Arc Three Point--
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA1000,100;PD2500,100;";
70 LPRINT "PU650,1150;PD1000,1150;";
80 LPRINT "PU650,450;PD1000,450;";
90 LPRINT "PU1000,100;PD1000,1500,2500,1500;";
100 LPRINT "AT3200,800,2500,1500;";
110 LPRINT "PU3200,900;PD;";
120 LPRINT "AT3300,800,3200,700;";
130 LPRINT "PU3300,800;PD3500,800;";
140 LPRINT CHR$(27); "%0A";
150 LPRINT CHR$(27); "E";
160 END

```

<Sample 28>

RT - Relative arc three point

RT X_1, Y_1, X_2, Y_2 [*qd*] [;]

X_1 ; x-coordinate of intermediate point

Y_2 ; y-coordinate of intermediate point

X_2 ; x-coordinate of end point

Y_2 ; y-coordinate of end point

qd; the chord angle used to draw the arc

- The specified coordinates of the two points are relative to the current pen position.
- The command draws an arc from the current position, through the specified intermediate position and finishing at the specified end position.
- After completion of the command the current cursor position is at the end of the arc.
- An arc can be drawn clockwise or anti-clockwise.
- If the current scaling mode is not isotropic, the arc may be elliptical rather than circular.
- The chord angle must be in the range 0.5° to 180°.
- The default chord angle is 5°.

```

10 '--Relative Arc Three Point--
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "&0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA1000,100;PR;PD1500,0;";
70 LPRINT "PU-1850,1050;PD350,0;";
80 LPRINT "PU-350,-700;PD350;0;";
90 LPRINT "PU0,-350;PD0,1500,1500,0;";
100 LPRINT "RT800,-750,0,-1500;";
110 LPRINT "PU700,850;PD;";
120 LPRINT "RT100,-100,0,-200;";
130 LPRINT "PU100,100;PD200,0;";
140 LPRINT CHR&(27); "%0A";
150 LPRINT CHR$(27); "E";
160 END

```

<Sample 29>

PE - Polyline encoded

PE [[*flag*] [*value*] | X_iY_i ... [*flag*] [*value*] | X_iY_i]];

flag ; a command or mode that applies to the data that follows

value; parameter data for the preceding flag

X ; X-coordinate of a cursor movement destination

Y ; Y-coordinate of a cursor movement destination

- This command allows you to combine a sequence of PA, PR, PU, PD and SP commands into a coded format. This reduces file size and transmission time.
- Do not use commas within the PE command. Parameters either have their own terminators or do not require them.
- The PE command must be terminated with a semi-colon.
- The command draws lines using the current line type and current units.
- The command draws lines to all coordinate points specified unless you precede the coordinate pair with a pen up flag.
- On completion of the command, the pen is down unless the final move was made with the pen up.
- All coordinate points are interpreted as relative unless preceded by the absolute flag (=).
- On completion of the command, the previous plotting mode is restored.
- Flags are not encoded. They are sent as ASCII characters. The MSB of the code is ignored, thus 60 and 188 can both represent <, the flag for pen up.

- The following *flags* are used:
 - '!' - Select pen. The number which follows is the required pen number. A PE command that does not include a select pen command uses the currently selected pen.
 - '<' - Pen up. The pen is raised and moved to the specified coordinate pair. All coordinate pair values not preceded by this flag are automatically interpreted as pen down plotting commands.
 - '>' - Fractional data. The value following the flag specifies the number of fractional binary bits in the coordinate data.
 - '=' - Absolute plotting mode. The pair of coordinates which follow this flag are absolute coordinates.
 - '7' - 7 bit mode. All subsequent coordinate values within this PE command are to be interpreted as 7-bit values, that is, encoded in base 32.
- If you use the '!' flag in polygon mode it is ignored as the SP command has no effect in this mode.
- Values and coordinates are encoded in base 64 or base 32. Values determine the setting of the immediately preceding flag. Legal values and coordinates are as follows:

Pen number -	0 (white) or 1(black)
Number of fractional binary bits -	-26 to 26. The default is 0.
x- and y-coordinates -	(-2^{30}) to $2^{30} - 1$ current units. If the pen position is moved outside this range, subsequent plotting commands are ignored until an absolute coordinate pair within the allowable range is specified.
- Flag and coordinate values are encoded as either base 64 or base 32 numbers and then transmitted as ASCII character codes. Base 64 is the default. Use Base 64 if your computer can send data without a parity bit. Use base 32 if your system requires a parity bit.
- To encode an integer proceed as follows. If the number is negative, take the absolute value, multiply by 2 and add 1. Hence $-x := 2x+1$. If the number is positive simply multiply by 2. Hence $x := 2x$. Convert the new number into base 64 or base 32 according to your system and encode each base 64 or base 32 digit as the corresponding ASCII character.
- To encode a real number proceed as follows. Multiply the number of decimal places in your coordinate's data by 3.33 and round the result up to the next highest integer (for example round 6.66 up to 7). This gives the number of binary bits needed to represent the number's fractional part - the value that you will supply with the '>' flag. Call this number n. Now multiply the number you are encoding by 2^n . Round this number to the nearest integer and then follow the procedure described above for encoding an integer.
- Transmit each number to the printer least significant digits first. Terminate each number with the most significant digit. This must be specified from a different ASCII range from the preceding digits in the number. In base 64, non-terminating digits are represented by the numbers 63-126 and terminating digits by the numbers 191-254. In base 32, non-terminating digits are represented by the numbers 63-94 and terminating digits by the numbers 95-126. Hence if using a base 32 number whose least significant digit is 14, and whose most significant digit is 5, encode 14 as 77 (63+14) and 5 as 100 (95+5).

	Non-terminator	Terminator
Base 64	63-126	191-254
Base 32	63-94	95-126

- In symbol mode the PE command draws the specified symbol at each specified point.
- In polygon mode the points specified within the PE command are not plotted. Instead they are stored in the polygon buffer and used when a FP (Fill Polygon) or EP (Edge Polygon) command is used.
- The PE command with no parameters simply updates the carriage return point.

CI - Circle plot

CI *r* (, *qd*) [;]

r ; Radius of circle (in current units)

qd ; Chord angle (in degrees)

- The command plots a circle with the current position as the centre, with a radius *r* and chord angle *qd*.
- After plotting, the cursor returns to the centre of the circle.
- Plotting takes place irrespective of whether the pen is up or down.
- Valid values for *r* are specified in the current unit.
- Valid values for *qd* are clamped real numbers in the range 0.5° to 180° . The default value is 5° .

- To avoid placing an unwanted dot at the centre of the circle precede the CI command with a PU command.

```

10 '-- Circle --
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27);"E";
40 LPRINT CHR$(27);"%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "SC-75,75,-75,75,1;"
70 LPRINT "PA0,0;LT;CI5;LT0;CI-12;"
80 LPRINT "LT1;CI19;LT2;CI-26;"
90 LPRINT "LT3;CI33;LT4;CI-40;"
100 LPRINT "LT5;CI47;LT6;CI54;"
110 LPRINT CHR$(27);"%0A";
120 LPRINT CHR$(27); "E";
130 END

```

<Sample 30>

BR - Bezier relative

BR [X1, Y1, X2, Y2, X3, Y3 . . . [X1, Y1, X2, Y2, X3, Y3]] [;]

X1, X2, X3 ; X coordinate of the second (X1), third (X2), and forth (X3) control point

Y1, Y2, Y3 ; Y coordinate of the second (Y1), third (Y2), and forth (Y3) control point

- Coordinates are relative values in current units.
- Using the current pen position as the starting point, the bezier curve is geometrically drawn referring to the other three points. The start point of the curve is at (X0,Y0), and the curve is tangential to the line from (X0, Y0) to (X1, Y1) at that point, and it leaves the point in that direction. The end point of the curve is at (X3,Y3), and the curve is tangential to the line from (X2,Y2) to (X3,Y3).
- After execution of this command the cursor position is at the end of the bezier curve.
- When drawing a bezier curve is repeated, the last (fourth) control point of the first curve is used as the first control point for the second curve.
- Plotting takes place only when the pen is down.
- If the command is invoked with the pen up, plotting is not performed, but the cursor position moves to what would have been the plot end point.
- If the curve extends to the outside of the soft/hard clip limit, it is clipped.
- This command can only be used in polygon mode.

BZ - Bezier absolute

BZ [X1, Y1, X2, Y2, X3, Y3 . . . [X1, Y1, X2, Y2, X3, Y3]] [;]

X1, X2, X3 ; X coordinate of the second (X1), third (X2), and forth (X3) control point

Y1, Y2, Y3 ; Y coordinate of the second (Y1), third (Y2), and forth (Y3) control point

- Coordinates are absolute values in current units.
- Using the current pen position as the starting point, the bezier curve is geometrically drawn referring to the other three points. The start point of the curve is at (X0,Y0), and the curve is tangential to the line from (X0, Y0) to (X1, Y1) at that point, and it leaves the point in that direction. The end point of the curve is at (X3,Y3), and the curve is tangential to the line from (X2,Y2) to (X3,Y3).
- After execution of this command the cursor position is at the end of the bezier curve.
- When drawing a bezier curve is repeated, the last (fourth) control point of the first curve is used as the first control point of the second curve.
- Plotting takes place only when the pen is down.
- If the command is invoked with the pen up, plotting is not performed, but the cursor position moves to what would have been the plot end point.
- If the curve extends to the outside of the soft/hard clip limit, it is clipped.
- This command can only be used in polygon mode.

7.4. The polygon group

Instruction	Function
PM	Polygon Mode
EA	Edge Absolute Rectangle
ER	Edge Relative Rectangle
EP	Edge Polygon
EW	Edge Wedge
RA	Fill Absolute Rectangle
RR	Fill Relative Rectangle
WG	Fill Wedge
FP	Fill Polygon

PM - Polygon mode

PM [*ms*] [;]

- This command is used for entering and leaving polygon mode, the mode that allows you to access the polygon buffer.
- When in polygon mode you can use any of the vector group commands to define the vertices of a polygon. The following commands can be used;
DF, IN, AA, AR, AT, BR, BZ, CI, PA, PD, PE, PR, PU, RT
- The polygon you have defined is not printed until you exit polygon mode and fill or edge the polygon.
- *ms*, the mode status, can have the value 0, 1 or 2. If *ms* = 0 the command empties the polygon buffer and enters polygon mode. A value of 1 closes the current polygon and remains in polygon mode. All commands sent following a PM1 before a PM2 or a further PM1 are used to build one sub-polygon. A value of 2 closes the current polygon or sub-polygon and exits polygon mode.
- Having exited polygon mode you can use the EP and FP commands to edge or fill the polygons you have defined.
- The EP command only draws between points defined when the pen was in the down position.
- The FP command fills the area between the polygon vertices irrespective of whether the pen was up or down when the polygon was defined.
- The first point in the polygon buffer is the pen position when the PM0 command was invoked.
- The next pen position specified after a PM1 command becomes the first vertex of the next sub-polygon.
- When plotting a polygon the pen always moves to the first point in the up position.
- Using the command with no parameters clears the polygon buffer and enters polygon mode. This is equivalent to the PM0; command.
- If a DF or IN command is invoked in the polygon mode, the printer quits polygon mode, clears the polygon buffer and executes the subsequent command.
- If an ESC E reset command is invoked while the printer is in polygon mode, the printer quits polygon mode, clears the polygon buffer, exits HP-GL/2 and ejects the page.

EA - Edge rectangle absolute

EA X, Y [;]

X ; X coordinate of opposite corner for the rectangle
Y ; Y coordinate of opposite corner for the rectangle

- X and Y coordinates are absolute coordinates in current units.
- Plots the rectangle formed by the current position and the opposite corner specified by X and Y.
- After plotting, the cursor returns to its point of origin and pen up/down status remains as selected.
- Plotting is performed whether the pen is up or down.

```

10 '-Edge Rectangle Absolute-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "SC0,150,0,150,1;";
70 LPRINT "PA25;105;EA65;130;";
80 LPRINT "PA30;110;EA70,135;";
90 LPRINT "PA35;115;EA75;140;";
100 LPRINT CHR$(27); "%0A";
110 LPRINT CHR$(27); "E";
120 END

```

<sample 31>

ER - Edge rectangle relative

ER X, Y [;]

X ; X coordinate of opposite corner for the rectangle

Y ; Y coordinate of opposite corner for the rectangle

- Coordinates are relative to the current position in current units.
- Plots the rectangle formed by the current position and the opposite corner specified by X and Y.
- After plotting, the cursor returns to its point of origin.
- Plotting is performed whether the pen is up or down.

```

10 '-Edge Rectangle Relative -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "SC0;150;0;150;1;";
70 LPRINT "PA25,105;ER40,25;";
80 LPRINT "PR 5, 5;ER40,25;";
90 LPRINT "PR 5, 5;ER40,25;";
100 LPRINT CHR$(27); "%0A";
110 LPRINT CHR$(27); "E";
120 END

```

<Sample 32>

EP - Edge polygon

EP [;]

- This command plots the outline of the polygon currently stored in the polygon buffer.
- The command only plots between points defined when the pen was down.
- The command plots an outline irrespective of whether the pen is currently up or down.
- The command does not affect the data in the polygon buffer.
- On completion of the plot the cursor returns to the original position.

EW - Edge wedge

EW r, q1, qc (, qd) [;]

r ; Radius in current units

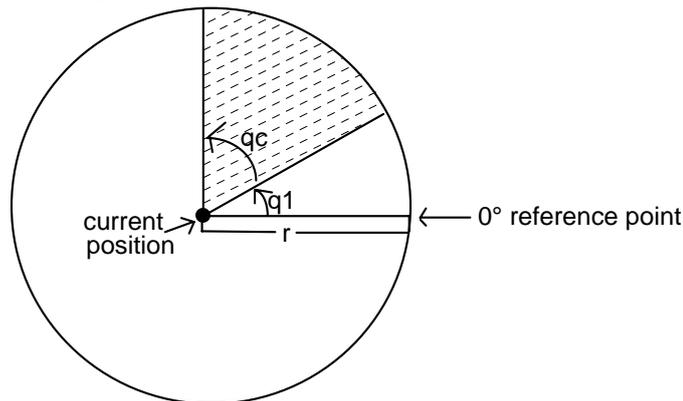
q1 ; Start point angle

qc ; Arc angle

qd ; Chord angle

- Plots a wedge centred on the current position with radius r , start point angle $q1$, arc angle qc and chord angle qd .
- After plotting, the cursor returns to its point of origin.
- Plotting is performed whether the pen is up or down.
- You should use this command with the type "isotropic". Anisotropic type may make the wedge stretch.

- r specifies the radius of the circle. A positive value for r places the 0° reference point onto the positive x-axis and a negative value for r places the 0° reference point onto the negative x-axis.
- The value for $q1$ is a clamped real number.



- $q1$ specifies the wedge starting point relative to the 0 degree reference point.
- When $q1$ is positive, the positive direction of the X axis relative to the current position is set at 0 degrees and the start point is sought in the counterclockwise direction. The opposite occurs when $q1$ is negative: the negative X axis is set at 0 degrees and the start point is sought by going clockwise.
- qc specifies the angle of the wedge in degrees.
- The value for qc is a clamped real number.
- Plotting proceeds counterclockwise when qc is positive and clockwise when negative.
- The value for qd must be an integer from -32,768 to +32,767.
- When qd is not specified, the chord angle is the default value (5 degrees)

```

10 '- Edge Wedge -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA2000,5000;";
70 LPRINT "EW1250,90,180,5;";
80 LPRINT "EW1250,270,120;";
90 LPRINT "EW1250,30,60;";
100 LPRINT CHR$(27); "%0A";
110 LPRINT CHR$(27); "E";
120 END

```

<Sample 33>

RA - Fill rectangle absolute

RA X, Y[;]

X ; X coordinate of opposite corner for the rectangle

Y ; Y coordinate of opposite corner for the rectangle

- X and Y coordinates are absolute coordinates in current units.
- Fill the rectangle formed by the current position and the opposite corner specified by X and Y.
- After plotting, the cursor returns to its point of origin.
- Plotting is performed whether the pen is up or down.

```

10 ' - Fill Absolute Rectangle -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;PA5000,4000;";
60 LPRINT "FT1;RA4250,3250;";
70 LPRINT "FT3,100;RA5750,3250;";
80 LPRINT "FT2;RA5750,4750;";

```

```

90 LPRINT "FT4,100,45;RA4250,4750;"
100 LPRINT CHR$(27); "%0A";
110 LPRINT CHR$(27); "E";
120 END

```

<Sample 34>

RR - Fill rectangle relative

RR *x*, *y* [;]

X ; X coordinate of opposite corner for the rectangle

Y ; Y coordinate of opposite corner for the rectangle

- Coordinates relative to the current position in current units.
- Fill the rectangle formed by the current position and the opposite corner specified by *X* and *Y*.
- After plotting the cursor returns to its point of origin.
- Plotting is performed whether the pen is up or down.

```

10 ' - Fill Relative Rectangle -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;PA5000,4000;"
60 LPRINT "FT1;RR750,750;"
70 LPRINT "PR750,0;"
80 LPRINT "FT3,100;RR750,750;"
90 LPRINT "PR0,750;"
100 LPRINT "FT2;RR750,750;"
110 LPRINT "FT4,100,45;RR-750,750;"
120 LPRINT CHR$(27); "%0A";
130 LPRINT CHR$(27); "E";
140 END

```

<Sample 35>

WG - Fill wedge

WG *r*, *q1*, *qc* (, *qd*) [;]

r ; Radius in current units

q1 ; Start point angle

qc ; Arc angle

qd ; Chord angle

- Fill in a wedge centred on the current position with radius *r*, start point angle *q1*, arc angle *qc* and chord angle *qd*.
- After plotting, the cursor returns to its point of origin.
- Plotting is performed whether the pen is up or down.
- You should use this command with the type "isotropic". Anisotropic type may make the wedge stretch.
- *r* specifies the radius of circle. A positive value places the 0° reference point onto the positive x-axis and the negative value places it onto the negative x-axis.
- The value for *q1* is a clamped real number.
- *q1* specifies the wedge starting point related to the 0° reference point.
- When *q1* is positive, the positive direction of the X axis relative to the current position is set at 0 degrees and the start point is sought in the counterclockwise direction. The opposite occurs when *q1* is negative: the negative X axis is set at 0 degrees and the start point is sought by going clockwise.
- *qc* specifies the angle of the wedge.
- The value for *qc* is a clamped real number.
- Plotting proceeds counterclockwise when *qc* is positive and clockwise when negative.
- The value for *qd* is a clamped real number.
- When *qd* is not specified, the chord angle is the default value (5 degrees).

```

10 ' - Fill Wedge -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP2;FT3,100;"
60 LPRINT "PA2000,5000;"
70 LPRINT "WG1250,90,180,5;"
80 LPRINT "SP3;EW1250,90,180,5;"
90 LPRINT "SP4;FT4,100,45;"
100 LPRINT "WG1250,270,120;"
110 LPRINT "SP3;EW1250,270,120;"
120 LPRINT "SP1;FT1;"
130 LPRINT "WG1250,30,60;"
140 LPRINT "SP3;EW1250,30,60;"
150 LPRINT CHR$(27); "%0A";
160 LPRINT CHR$(27); "E";
170 END

```

<Sample 36>

FP - Fill polygon

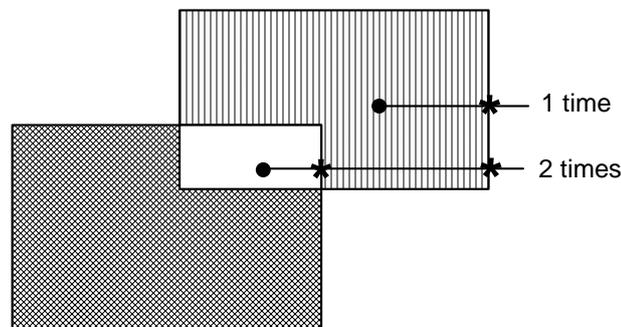
FP [;]

FP [*n*] [;]

n ; fill method

- This command fills the polygon currently stored in the polygon buffer.
- The command does not clear the buffer or alter the data at all.
- The command fills between points defined when the pen was up or down.
- If a polygon is composed of sub-polygons, the command fills alternate areas.
- On completion of the plot the cursor returns to the original position.
- The command performs a fill irrespective of whether the pen is currently up or down.
- You can specify the fill method. There are two types of fill method.
- 0 specifies the even/odd fill method.

Places a point within the polygon and draws a straight line from it to the outside of polygon. If the cross point of the straight line and polygon edges are an odd count, the polygon which includes the start point is filled.



- 1 specifies Non-zero winding fill method.

```

10 ' - Fill Polygon -
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA1500,1500;PM0;CI1000,60;PM1;"
70 LPRINT "CI500;PM1;PM2;"
80 LPRINT "LT4;FT3,50,45;"
90 LPRINT "FP;"
100 LPRINT CHR$(27); "%0A";
110 LPRINT CHR$(27); "E";
120 END

```

<Sample 37>

7.5. Plot Function Instructions

Instruction	Function
AC	Anchor Corner
FT	Fill Type
LA	Line Attribute
LT	Line Type
PW	Pen Width
RF	Raster Fill Definition
SM	Symbol Mode
SP	Select Pen
SV	Screened Vectors
TR	Transparency Mode
UL	User-defined Line Type
WU	Select Unit for Pen Width

AC - Anchor corner

AC [X, Y] [;]

X ; x-coordinate of starting point of fill pattern

Y ; y-coordinate of starting point of fill pattern

- This command specifies the anchor corner which is the starting position of a fill pattern.
- If no parameters are specified the lower left hand corner of the PCL picture frame, according to the current coordinate system, becomes the anchor position. This is equivalent to AC0,0;
- User units or graphics units may be used.
- The value of the coordinates are real numbers in the range (-2^{30}) to $2^{30} - 1$.

```

10 '-Anchor Corner -
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA3000,3000;FT3,400,45;";
70 LPRINT "RR1000,1000;ER1000,1000;";
80 LPRINT "PR1000,0;FT4,400,45;";
90 LPRINT "RR1000,1000;ER1000,1000;";
100 LPRINT "PR1000,0;FT3,400,45;";
110 LPRINT "RR1000,1000;ER1000,1000;";
120 LPRINT "PA3000,1500;AC3000,1500;";
130 LPRINT "RR1000,1000;ER1000,1000;";
140 LPRINT "PA4000,1500;AC4000,1500;FT4,400,45;";
150 LPRINT "RR1000,1000;ER1000,1000;";
160 LPRINT "PA5000,1500;AC5000,1500;FT3,400,45;";
170 LPRINT "RR1000,1000;ER1000,1000;";
180 LPRINT CHR$(27); "%0A";
190 LPRINT CHR$(27); "E";
200 END

```

<Sample 38>

FT - Fill type**FT** *n*(, *d*(, *q*)) [;]*n* ; Fill type *d* ; option 1 *q* ; option 2

- Sets the fill type, interval and angle when filling an area. This setting is effective for the commands FP, RA, RR, WG and CF.
- The value for *n* must be a clamped integer number

Fill type	Description	Option 1	Option 2
1	Solid black (bi-directional fill)		
2	Solid black (single directional fill)		
3	hatching (parallel line)	fill interval	fill angle
4	cross-hatching	fill interval	fill angle
10	shading	shading level	
11	HP-GL/2 user-defined	raster-fill index	
21	PCL cross-hatching	pattern type	
22	PCL user-defined	pattern ID	
130	shading (one of sixty-four gray scales)	Shading level	

- option 1 for *n*=3 or 4 is the setting for the interval of the lines in the fill. This is specified in current units.
- If *d* is omitted or 0, it is set to the default value which is 1% of the diagonal distance between P1 and P2.
- Option 2 for *n*=3 and 4 is the setting of the angle for the lines in degrees.
- If *q* is omitted, the fill angle already specified will be used.
- The value of *q* is a clamped real number.
- Option 1 for *n*=10 is the setting for the shading density. It is specified as a percentage from 0 to 100.
- Option 2 for *n*=10 is ignored.
- Option 1 for *n*=11 specifies the HP-GL/2 user-defined raster fill pattern. This uses the index number which is defined by the RF command.
- Option 2 for *n*=11 is ignored.
- Option 1 for *n*=21 specifies the PCL cross-hatch patterns. There are 6 types of PCL cross-hatch patterns.
- Option 2 for *n*=21 is ignored.
- Option 1 for *n*=22, the PCL user-defined patterns defined by the ESC*c#W command with a unique ID code.
- Option 2 for *n*=22 is ignored.
- Option 1 for *n*=130, the 64 gray scale shading density. It is specified as a percentage from 0 to 100.
- Option 2 for *n*=130 is ignored.

```

10 ' - Fill Type -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA2000,2000;FT;PR2500,300;ER2500,300;";
70 LPRINT "PR0,300;FT3,80,30;RR2500.300;";
80 LPRINT "PR0,300;FT10,36;RR2500,300;ER2500,300;";
90 LPRINT CHR$(27); "%0A";
100 LPRINT CHR$(27); "E";
110 END

```

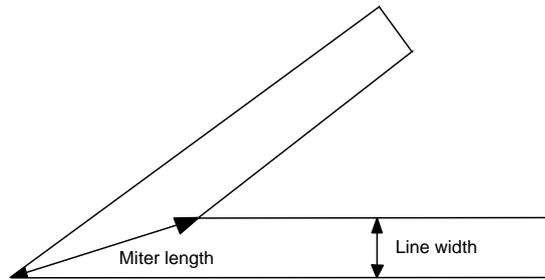
<Sample 39>

LA - Line attribute

LA [*l*, *v* [, *l*, *v* [, *l*, *v*]]] [;]

l : line attribute *v* : value

- This command specifies the shape of line joins and line ends.
- These settings are effective for the line widths thicker than 0.35 mm.
- *l* specifies the line attribute for which you are setting a value.
- There are three line attributes - line end type, line join type and miter limit, specified by line attribute values of 1, 2 and 3 respectively.
- *v* specifies the value of the particular attribute.
- Line ends are either butt (1), square (2), triangular (3) or round (4).
- Line joins are either mitered (1), mitered/beveled (2), triangular (3), round (4), beveled (5), or have no join (6).
- Miter limit is the ratio of miter length to line width. When the miter limit is exceeded, the join is clipped to the miter limit.
- The miter limit is specified as a clamped real number.
- If no parameters are specified, the line attributes are set as follows: line ends are butt, line joins are mitered and the miter limit is 5. This is equivalent to the command LA1, 1, 2, 1, 3, 5;



$$\text{Miter limit} = \text{Miter length} / \text{Line width}$$

- Labels are always drawn with rounded ends and joins.
- An LA command remains in effect until another LA command is issued, until default settings are restored with the DF command or until the printer is initialized with an IN command.

```

10 '-Line Attributes-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA4000,3000;PW2;LA1,3;PD3500,2500,4000,2000;";
70 LPRINT "PU3500,200;LA2,2,3,20;PD3000,2500,3000,2300;";
80 LPRINT "PU2500,2300;LA1,4;PD3500,2300;";
90 LPRINT "PU2700,2100;PD3300,2100;";
100 LPRINT "PU2900,1900;PD3100,1900;";
110 LPRINT CHR$(27); "%0A";
120 LPRINT CHR$(27); "E";
130 END

```

<Sample 40>

LT - Line type selection

LT [*n* [, *p* [, *m*]]] [;]

n ; Line pattern number

p ; Line pattern length (percentage or millimeters of distance between P1 and P2)

m : mode

- This command specifies the line pattern for use when drawing lines with the vector group commands.
- *n* defines the line type to be used. *n* is either in the range -8 to 8 or is 99.
- Values of *n* from 1 to 8 specify a line type (as defined with the UL command) and use a fixed pattern length. Any residual part of the pattern not used on a particular drawing operation is stored and used at the start of the next line drawn.
- Values of *n* from -1 to -8 specify a line type (as defined with the UL command) and use an adaptive pattern length - that is, the pattern is scaled so as to fit one or more complete patterns into the line.
- If *n* is 0, a dot is placed at the (X, Y) coordinates specified with any vector commands used.
- *p*, the pattern length, specifies the length of one complete pattern, either as a percentage of the distance between P1 and P2 or in millimeters. The unit system used is determined by the mode setting. If the length is omitted, the printer uses the most recently specified pattern length.
- If a zero or negative pattern length is specified the command is ignored.
- *m*, the mode, defines whether the pattern length parameter, *p*, is interpreted as a reading in millimeters, (*m*=1), or as a percentage of the distance between P1 and P2, (*m*=0). If no mode number is included in the command, the most recently specified mode is used.
- LT99; restores the previous line type (and, if it is a fixed line type, the residue), provided that the current line type is solid and that the pen position is the same as it was when the current, solid line type was selected.
- An LT command remains in effect until another LT command is issued, until the default settings are restored with the DF command or until the printer is initialized with an IN command.
- If no parameters are specified the line type is set to be solid. The previous line type, pattern length and any line residue are saved.

```

10 ' - Line Type -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;IP1000,1000,9000,7000;SP1;";
60 LPRINT "PA2000,7100;PDPR400,0;PU;" 'REM DEFAULT LINE = 400
70 LPRINT "LT8;PA2000,7000;PD5000,7000;PU;"
80 LPRINT "LT8,4,0;PA2000,6900;PD5000,6900;PU;"
90 LPRINT "LT8,8,1;PA2000,6800;PD5000L6800;PU;"
100 '
110 LPRINT "LT4,PA2000,6600;PD5000,6600;PU;"
120 LPRINT "LT4,4,0;PA2000;6500;PD5000;6600;PU;"
130 LPRINT "LT4,10,1;PA2000,6400;PD5000,6400;PU;"
140 '
150 LPRINT "LT0;PA2000,6300;PD5000,6200;PU;"
160 LPRINT "LT0,4,0;PA2000,6100;PD5000,6100;PU;"
170 LPRINT "LT0,10,1;PA2000,6000;PD5000,6000;PU;"
180 '
190 LPRINT "LT-4;PA2000,5800;PD5000,5800;PU;"
200 LPRINT "LT-4,4,0;PA2000,5700;PD5000;5700;PU;"
210 LPRINT "LT-4,10,1;PA2000,5600;PD5000,5600;PU;"
220 '
230 LPRINT "LT-8;PA2000,5400;PD5000,5400;PU;"
240 LPRINT "LT-8,4,0;PA2000,5300;PD5000,5300;PU;"
250 LPRINT "LT08,10,1;PA2000,5200;PD5000,5200;PU;"
260 '
270 LPRINT "LT99;PA2000,5000;PD5000,5000;PU;"
280 LPRINT CHR$(27); "%0A";
290 LPRINT CHR$(27); "E";
300 END

```

<Sample 41>

PW - Pen width**PW** [*w*, [, *p*]] [;]*w* ; width*p* ; pen

- This command specifies the width of the pen.
- Pen width can either be specified as a fixed width in millimeters (the default) or relative to the distance between the scaling points P1 and P2.
- If a width of zero is specified the thinnest width possible (1 dot wide) is set.
- Metric widths are scaled by the ratio of the PCL picture frame to the HP-GL/2 plot size. If the ratio for the two axes is different, the smaller one is used. If this is less than the thinnest width available then a width of 1 dot is selected.
- If no pen number is specified the new width applies to both pens.
- If a pen number other than 0 or 1 is specified the command is ignored. There are six pens in the specification.
- If no parameters are specified the pen width defaults to either 0.35mm or 0.1% of the distance from P1 to P2 according to the current width unit setting.
- Pen width settings are not affected by a DF command.
- Pen width does not set the width of the lines that make up the characters in a label.
- A PW command remains in effect until another PW command is issued, until a WU command is used or until the printer is initialized with an IN command.

```

10 '-Pen Width-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA3500,2500;PW1.5;";
70 LPRINT "PD4500,2800,4500,1800,3500,1500,3500,2500;";
80 LPRINT "PW0.8;PD2300,2900,2300,1900,3500,1500;";
90 LPRINT "PW0.5;PU2300,2900;PD3300,3200,4500,2800;";
100 LPRINT "PW0.25;PU4500,1800;PD3500,2100;";
110 LPRINT CHR$(27); "%0A";
120 LPRINT CHR$(27); "E";
130 END

```

<Sample 42>

RF - Raster fill definition**RF** [*i* [, *w*, *h*, *p* [, ...]]] [;]*i* : index*w* : width*p* : pen number*h* : height

- This command defines a rectangular pattern which may be used to fill the area.
- *i* is the index number of the pattern that is to be defined and must be in the range 1-8. 8 patterns can exist concurrently.
- If used with an index parameter and no others, the pattern is defined as a solid black fill.
- *w* is the width in pixels of the pattern that is to be defined and must be in the range 1- 255.
- *h* is the height in pixels of the pattern that is to be defined and must be in the range 1 - 255.
- *p* stands for a pixel in the defined fill. *p* = 0 signifies a white pixel, *p* > 0 signifies a black pixel. The number of pixels specified should equal the number of pixels in the fill pattern (*w*, *h*). The parameter defines successive pixels' colors from left to right, top to bottom. If you do not supply enough pixel parameters the trailing pixels default to white.
- Use the FT command to select a fill type from those defined with the RF command.
- Omitting parameters sets all raster fill patterns to solid fill.


```

170 'Source 2
180 LPRINT "FT10,50;PA4000,5000;RR500,500;PA4500,5500;RR500,500;"
190 LPRINT "FT10,1;PA4500,5000;RR500,500;PA4000,5500;RR500,500;FT;"
200 '
210 '( POLYGON )
220 LPRINT "PA2000,1000;PM0;PDPA4000,1000,3000,3000;PM1;"
230 LPRINT "PA2500,1200,3500,1200,3000,2500;PM1;"
240 LPRINT "PA3000,800,3700,2500,2300,2500;PM1;"
250 LPRINT "PM2;FP;"
260 LPRINT CHR$(27); "%0A";
270 LPRINT CHR$(27); "E";
280 END

```

<Sample 45>

UL - User-defined line type

UL [*i*] [, *g* [...]] [;]

i ; unit type *g* : gap length

- This command allows you to define line types which you can then subsequently select with the LT command.
- *i* must be in the range 1-8. *i* identifies the line type that is to be defined.
- The index parameter uses absolute values.
- *g* (the gap length) alternately specifies pen up and pen down parts of the defined line type. A maximum of twenty line sections are permitted, the first of which is a pen down section. Numbers must be integers in the range 0~32767. These numbers are converted to percentages by the LT command.
- Maximum definition for the gap of each line type is 20.
- The sum of the gap parameters must be greater than 0.
- Specifying an index number with no gap parameters sets the line type to the default pattern for that index number.
- Using the command with no parameter sets all line types to the defaults.

```

10 '-User Defined Line Type-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "UL8,0,15,0,15,0,15,40,15;"
70 LPRINT "LT8,5;PA2000,2500;PDPA5000,2500;PU;"
80 LPRINT "UL4,40,15,0,15,0,15,40,15,0,15;"
90 LPRINT "LT4,10;PA2000,3000;PDPA5000,3000;PU;"
100 LPRINT "UL1,40,40,40,40,40;"
110 LPRINT "LT1,10;PA2000,3500;PDPA5000,3500;PU;"
120 LPRINT CHR$(27); "%0A";
130 LPRINT CHR$(27); "E";
140 END

```

<Sample 46>

WU - Select unit for pen width

WU [*t*] [;]

t ; unit type

- This command specifies the pen width unit. It applies to all pens.
- *t* must be either 0 or 1.
- If *t* is 0 the pen width parameter which accompanies subsequent PW commands will be treated as a reading in millimeters.
- If *t* is 1 the pen width parameter which accompanies subsequent PW commands will be treated as percentage of the distance between the scaling points P1 and P2. The default width is 0.1% of the diagonal distance from P1 to P2.

- The command sets all pen widths to a default of 0.35mm. Hence it should be used before the PW command.
- Using the command with no parameter defaults the type to 0 (metric) and all pen widths to 0.35mm.
- The unit type is not affected by a DF command.

```

10 '-Pen Width Unit Selection-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 'metric
70 LPRINT PA2000,6000;PDPA6000,6000;PU;" 'Default 0.35mm
80 LPRINT "WU0;PA2000,5700;PDPA6000,5700;PU;" '0.35mm
90 LPRINT "WU0;PW1.5;PA2000,5400;PDPA6000,5400;PU;" '1.5mm
100 LPRINT "WU0;PW0.8;PA2000,5100;PDPA6000,5100;PU;" '0.8mm
110 "-relative unit
120 LPRINT "PW;PA2000,4600;PDPA6000,4600;PU;" 'Default 0.35mm=4dot
130 LPRINT "IP1000,1000,9000,7000;";
140 LPRINT "WU1;PA2000,4300;PDPA6000,4300;PU;" '300*10/1016=2dot
150 LPRINT "IP1000,1000,5000,4000;";
160 LPRINT "WU1;PA2000,4000;PDPA6000,4000;PU;" '300*5/1016=1dot
170 LPRINT CHR$(27); "%0A";
180 LPRINT CHR$(27); "E";
190 END

```

<Sample 47>

7.6. Character Plot Instructions

Instruction	Function
SD	Define Standard Font
AD	Define Alternate Font
SS	Select Standard Font
SA	Select Alternate Font
FI	Select Primary Font
FN	Select Secondary Font
LB	Define Label
DT	Define Label Terminator
LO	Label Origin
DI	Absolute Direction
DR	Relative Direction
DV	Define Variable Text Path
CP	Character Plot
CF	Character Fill Mode
SI	Set Absolute Character Size
SR	Set Relative Character Size
SL	Set Character Slant
SB	Scalable or Bitmap Fonts
ES	Extra Space
TD	Transparent Data

7.6.1. Introduction

The character group commands enable you to print text to accompany your HP-GL/2 generated graphics without leaving graphics mode. You can use any font that would be available to you in LaserJet emulation mode and you can modify various printing characteristics of the text such as the size, direction and slant of the characters.

SD - Define standard font

SD [*attribute*, *value*] [..] [;]

attribute ; a font attribute to be defined

value ; the value of the attribute

- This command allows you to define the standard font and its attribute.
- *attribute* takes a value from 1-7. 1=symbol set, 2=spacing, 3=pitch, 4=height, 5=posture, 6=stroke weight and 7=typeface.
- *value* selects an option for the specified attribute.
- If the defined font is not available another font is selected that matches the defined font's specified attributes as closely as possible. The symbol set attribute has the highest priority for matching and typeface has the lowest.
- If you omit all parameters, the default standard font attributes are used. These are shown below.

Attribute	Meaning	Default value	Description
1	Symbol set	277	Roman-8
2	Spacing	0	Monospacing
3	Pitch	9	9 cpi
4	Height	11.5	11.5 point
5	Posture	0	Upright characters
6	Stroke weight	0	Medium
7	Typeface	48	Stick font

- attribute = 1 : Symbol set

Value	Name	ISO number	Value	Name	ISO number
1	Math - 7	-	2	Line Draw - 7	-
3	HP Large Characters	-	4	Norwegian v1	60
5	Roman Extensions	-	6	French v1	25
7	HP German	-	8	Hebrew - 7	-
9	Italian	15	11	JIS ASCII	14
12	Line Draw - 7	-	13	Math - 7	-
14	ECMA-94 Latin1 (8-bit version)	8859/1	15	OCR - A	-
16	APL (typewriter paired)	-	18	Cyrillic ASCII	-
19	Swedish for names	11	20	Thai - 8	-
21	ASCII	6	22	Arabic (MacKay's Version)	-
25	3 of 9 Barcode	-	26	Not used	-
36	Danish/Norwegian v2	61	37	United Kingdom	4
38	French v2	69	39	German	21
43	Katakana	13	44	HP Block Characters	-
45	Tech - 7	-	47	OCR - B	-
48	APL (bit paired)	-	50	Cyrillic	-
51	HP Spanish	-	53	Legal	-
57	Industrial 2 of 5 Barcode	-	75	Chinese	57
76	Tax Line Draw	-	78	ECMA-94 Latin 2	8859/2
79	OCR-M	-	83	Spanish	17
85	International Reference Version	2	89	Matrix 2 of 5 Barcode	-
114	PC Cyrillic	-	115	Swedish	10
147	Portuguese	16	153	Interleaved 2 of 5 Barcode	-
173	PS Math	-	174	ECMA-128 Latin 5	8859/9
179	Portuguese	84	181	HPL Language Set	-
185	CODABAR Barcode	-	202	Microsoft Publishing	-
205	Ventura Math	-	211	Spanish	85
217	MSI/Plessey Barcode	-	234	DeskTop	-
243	HP European Spanish	-	249	Code 11 Barcode	-
263	Greek - 8	-	264	Hebrew - 8	-
266	Document	-	267	Kana - 8	-
268	Line Draw 8	-	269	Math - 8	-
275	HP Latin Spanish	-	276	Turkish - 8	-
277	Roman - 8 (Default)	-	278	Arabic - 8	-
281	UPC/EAN Barcode	-	299	Korean - 8	-
309	Windows	-	330	PS Text	-
332	PS ITC Zapf Dingbats	-	334	ECMA-113/88 Latin/Cyrillic	8859/5.2
341	PC-8	-	373	PC-8 D/N	-
405	PC-850	-	428	ITC Zapf Dingbats Series 300	-
458	Ventura U.S.	-	501	Pi Font	-
505	USPS Zip	-	531	HP-GL Download	-
563	HP-GL Drafting	-	595	HP-GL Special Symbols	-

- attribute = 2 : Spacing

Value	Meaning
0	Monospacing
1	Proportional spacing

- attribute = 3 : Pitch
Specify pitch in characters per inch (dpi). The pitch value you specify can be in the range 0 to 32767.9999. Pitch only applies to monospaced fonts.

- attribute = 4 : Height
Specify height in points. One point is 1/72". The height value you specify can be in the range 0 to 32767.9999. Proportionally spaced fonts are scaled according to the height you select. Monospaced fonts ignore this setting value.

- attribute = 5 : Posture

value	Meaning
0	Upright
1	Italic
2	Alternate italic

- attribute = 6 : Stroke weight

Attribute	Meaning
-7	Ultra Thin
-6	Extra thin
-5	thin
-4	Extra light
-3	Light
-2	Demi light
-1	Semi light
0	Medium, Book or Text
1	Semi bold
2	Demi bold
3	Bold
4	Extra bold
5	Black
6	Extra black
7	Ultra black
9999	Stick font at current pen width*

* This attribute value is only effective if the stick font is the current font.

- attribute = 7 : Typeface

You can select any typefaces available in LaserJet mode, from the internal fonts, card/cartridge fonts or download fonts.

```

10 '-Standard Font Definition-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA2200,7000;SD2,0,4,11.5,5,0;";
70 LPRINT "DT#;LBlaser printer#"
80 LPRINT "PA2000,5000;SD2,1,4,20,5,1;";
90 LPRINT "LBlaser printer#"
100 LPRINT "PA1000,3000;SD2,1,4,50,5,1;";
110 LPRINT "LBLASER PRINTER#"
120 LPRINT CHR$(27); "E";
130 LPRINT CHR$(927); "%0A";
140 END

```

<Sample 48>

AD - Define alternate font

AD [*attribute*, *value*] [..] [;]

attribute : a font attribute to be defined

value : the value of the attribute

- This command allows you to define the alternate font and its attribute.
- *attribute* takes a value from 1-7.
1 = symbol set, 2 = spacing, 3 = pitch, 4 = height, 5 = posture, 6 = stroke weight and 7 = typeface.
- *value* selects an option for the specified attribute. For further explanation about attribute values see the previous SD command.
- If the defined font is not available, another font is selected that matches the defined font's specified attributes as closely as possible. The symbol set attribute has the highest priority for matching and typeface has the lowest.
- If you omit all parameters, the default alternate font attributes are used.

```

10 '-Define Alternate Font -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 '- Standard font -
70 LPRINT "PA1000,5000;";
80 LPRINT "LBStandard font"; CHR$(3);
90 ' - Alternate font -
100 LPRINT "AD2,1,4,24,7,4101;";
110 LPRINT "PA1000,4000;";
120 LPRINT "SA";
130 LPRINT "LBAlternate font"; CHR$(3);
140 LPRINT CHR$(27); "%0A";
150 LPRINT CHR$(27); "E";
160 END

```

<Sample 49>

SS - Select standard font

SS [;]

- This command allows you to select the standard font, as defined by the SD command. All subsequently printed labels will appear in the standard font.
- The SS command is equivalent to including the SI control code (ASCII 15) in a label.
- The default standard font is the HP-GL stick font.
- The font remains current until the printer is initialized with an IN command, until the default conditions are resorted with the DF command or until an SA command is used.

SA - Select alternate font

SA [;]

- This command allows you to select the alternate font, as defined by the AD command. All subsequently printed labels will appear in the alternate font
- The SA command is equivalent to including the SO control code (ASCII 14) in a label.
- The font remains current until the printer is initialized with an IN command, until the default conditions are restored with the DF command or until an SS command is used.

FI - Select primary font**FI font-ID** [;]

font-ID ; the identity number of any accessible font.

- This command allows you to select any accessible font, which has previously been given an identity number in LaserJet mode, as the primary (standard) font. All subsequently printed labels will appear in the primary font.
- For scalable fonts this command must be preceded by an SD command specifying the font's point size or pitch.
- When the FI command is issued the primary (standard) font's attributes are set to be those of the selected font (provided it is available).

```

10 '-Select Primary Font & Select standard Font -
20 WIDTH "LPT1:",255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "*c15D";
50 LPRINT CHR$(27); "(s1p18v0s3b52T";
60 LPRINT CHR$(27); "*c6F";
70 LPRINT CHR$(27); "%0B";
80 LPRINT "IN;SP1;";
90 LPRINT "PA1500,1500;DT#;LBLaser Printer"; CHR$(13);CHR$(10); "#"
100 LPRINT "SD4,18;FI15;LBLaser Printer#"
110 LPRINT CHR$(27); "%0A";
120 LPRINT CHR$(27); "E";
130 END

```

<Sample 50>

FN - Select secondary font**FN font-ID** [;]

font-ID : the identity number of any accessible font

- This command allows you to select any accessible font, which has previously been given an identity number in PCL mode, as the secondary (alternate) font. All subsequently printed labels will appear in the secondary font.
- For scalable fonts the FN command must be preceded by an AD command specifying the font's point size or pitch.
- When the FN command is issued the secondary (alternate) font's attributes are set to be those of the selected font (provided it is available).

LB - Character plot**LB** [*cs*] < terminator > [;]*cs* : character string

- This command enables you to print character strings.
- You must supply a terminator at the end of your string. You can either use the default terminator, ETX (ASCII 3), or define your own terminating character using the DT command.
- A label can include non-printing characters such as a carriage return or line feed.
- Printing takes place irrespective of whether the pen is currently up or down.
- The label is printed starting from the current cursor position and using any label origin setting that you have made with the LO command.

```

10 '-Label-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA2500,2500;DT*";
70 LPRINT "SD1,21,2,1,4,25,5,0,6,3,7,52;SS;";
80 LPRINT "LBLaser printer*"
90 LPRINT CHR$(27); "%0A";
100 LPRINT CHR$(27); "E";
110 END

```

<Sample 51>

DT - Define label terminator**DT** [*c* [, *m*]] [;]*c* : Character*m* : mode

- This command allows you to specify the character that terminates strings printed with the LB instruction.
- The character immediately following DT is taken to be the terminator. Hence, do not put a space between the DT and your chosen character. If you do, space will be interpreted as the terminator in all subsequent LB commands.
- *m*, the mode, determines whether the terminating character is printed as part of the label or not. If *m* = 0 the character is printed ; if *m* = 1 it is not.
- If you do not specify a value for *m* the terminating character is not printed.
- If you leave out the character parameter the default character ETX (ASCII 3) is used as the terminator.
- This setting is effective until the another DT is invoked or IN or DF is performed.

```

10 REM - Define Label Terminator -
20 WIDTH "LPT:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP2;";
60 LPRINT "IN;SP2;SC0,5000,0,5000;";
70 LPRINT "PA0,4500;LBDefault control character ETX"; CHR$(13);CHR$(3);
80 LPRINT "LBterminates by performing end-"; CHR$(13);CHR$(3);
90 LPRINT "LBof-text function.";CHR$(3);
100 LPRINT "PA0,3500;DT@;LBPrinting characters terminate,"; CHR$(13);
"@";
110 LPRINT "LBbut are also printed.@";
120 LPRINT "PA0,3000;DT"; CHR$(7); "LBcontrol characters terminate";
CHR$(13);chr(7);
130 LPRINT "LBand perform their function."; CHR$(7)
140 LPRINT CHR$(27); "%01";
150 LPRINT CHR$(27); "E";
160 END

```

<Sample 52>

LO - Label origin**LO** [*p*] [;]

- This instruction allows you to position labels in different ways relative to the current pen location.
- You can centre, left justify or right justify labels relative to the current pen location and print them on, above or below the current location.
- You can combine an offset distance, equal to 25% of the current font's point size, with any of the above origin positions.
- *p* must either be in the range 1 - 9, 11 - 10 and 21. The label origin position specified by each *p* parameter value is shown below. 21 sets the label origin to the same one in PCL.

P=1	P=4	P=7 .
P=2	P=5	P=8 .
P=3	P=6	P=9 .
.P=11	P=14 .	P=17 .
•P=12	P=15	P=18 .
•P=13	P=16	P=19 .

- When you use an LO instruction the carriage return point is set to be the new label origin. To send the pen back to a label's origin when you have finished printing it, include a carriage return between the last character of the label and the terminating character.
- If a label contains embedded carriage returns, each character string following a carriage return is printed starting from the label origin.
- If you omit the parameter the label origin is the bottom left hand corner of the first character's character cell - equivalent to LO11;
- The command remains in effect until the printer is initialized with an IN command, until the default conditions are restored with the DF command or until another LO command is used.

```

10 '-Label Origin-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP;";
60 LPRINT "SC-4000,4000,-5000,5000;SI0.17,0.26;";
70 LPRINT "PA0,500;PD-500,0,0,-500,500,0,0,500;DT#;";
80 LPRINT "CI10;LO4;LBCentred on point#"
90 LPRINT "PU-500,0;CI10;LO18;Lleft centre offset#"
100 LPRINT "PU0,-500;CI10;LO13;LRight offset from point#"
110 LPRINT "PA500,0;CI10;LO3;LBright hang from point#"
120 LPRINT CHR$(27); "%0A";
130 LPRINT CHR$(27); "E";
140 END

```

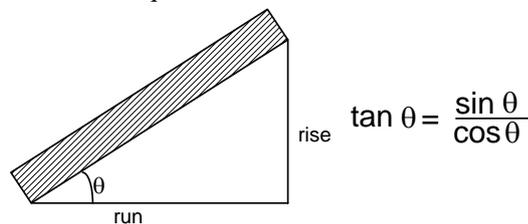
<Sample 53>

DI - Absolute direction

DI [*run*, *rise*] [;]

run : X direction component *rise* : Y direction component

- This command determines the direction in which subsequent labels will be printed.
- $\frac{\text{rise}}{\text{run}}$ gives the tangent of the angle between the base line of a label's characters and the horizontal.
- *run* is equivalent to $\cos \theta$ and *rise* is equivalent to $\sin \theta$.



- Values for *run* and *rise* are clamped real numbers.
- When you issue a DI instruction the carriage return point is set to the current pen location.
- The instruction is ignored if *run* and *rise* are both 0.
- Any text path that you set with the DV instruction will combine with the current printing direction to determine the layout of characters on the page.
- If you omit the parameters the printing direction is horizontal.
- The label direction which is specified by this command is not affected by changes in the location of P1 and P2.

```

10 '-Absolute Direction -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 DEG = -45
70 RAD = 3.1416 / 180 * DEG
80 LPRINT "IN;SP2;PA3050,4450;";

```

```

90 LPRINT "DT";"
100 LPRINT "DT0,2;LB*_1988#"; "DI2,2;LB*_1989#";
110 LPRINT "DT2,0;LB*_1990#"; "DI2,-2;LB*_1991#";
120 LPRINT "DT0,-2;LB*_1992#"; "DI-2,-2;LB*_1993#";
130 LPRINT "DT-2,0;LB*_1994#"; "DI-2,2;LB*_1995#";
140 LPRINT "PA3500,5350;DI", COS(0),SIN(0); "LB*_2000"; CHR$(13); "#";
150 LPRINT "DI", COS(RAD); SIN(RAD); "LB_RETURN_POINT"; CHR$(13); "#";
160 LPRINT CHR$(27); "%0A";
170 LPRINT CHR$(27); "E";
180 END

```

<sample 54>

DR - Relative direction

DR [*run*, *rise*] [;]

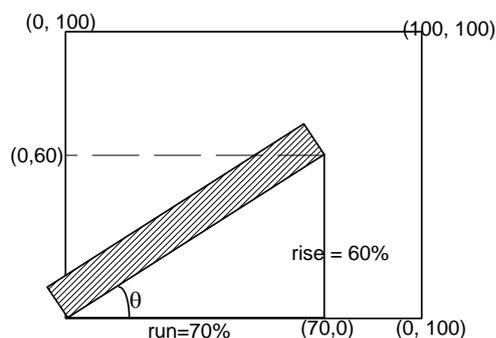
run : X-component of the printing direction expressed as a percentage of the x-component of the distance between P1 and P2.

rise : Y-component of the printing direction expressed as a percentage of the y-component of the distance between P1 and P2.

- This command determines the direction in which subsequent labels will be printed.
- Values for *run* and *rise* are clamped real numbers.
- The instruction is ignored if *run* and *rise* are both 0.
- When you issue a DR instruction the carriage return point is set to the current pen location.
- Any text path that you set with the DV instruction will combine with the current printing direction to determine the layout of characters on the page.
- If you change the positions of P1 and P2 the printing direction will change accordingly.
- If you omit the parameters the printing direction is horizontal and the *run* = 1% of |P2X-P1X|.
- Actual values for *run* and *rise* are changed relative to the P1 and P2 value. It is calculated as below:

$$\text{actual run} = \text{run parameter} \div 100 \times (\text{P2x} - \text{P1x})$$

$$\text{actual rise} = \text{rise parameter} \div 100 \times (\text{P2y} - \text{P1y})$$



DV - Define variable text path

DV [*path* [, *line*]] [;]

path : text printing path - horizontal or vertical, and left to right, right to left, upwards or downwards.

line: the effect of a line feed.

- This command determines the printing direction for labels and the carriage return point.
- Permissible values of *path* are 0, 1, 2 and 3. If *path* = 0 labels are printed horizontally and left to right - normal printing. If *path* = 1 labels are printed vertically downwards - characters appear in a vertical column. If *path* = 2 labels are printed horizontally and right to left - words appear backwards. If *path* = 3 labels are printed vertically upwards - characters appear in a vertical column, reading from bottom to top.
- The value of *line* must be either 0 or 1. If *line* = 0, each time a line feed is encountered the current pen position is moved at an angle of 90° to the path in a clockwise direction. Hence, if *path* = 1 the second line of text is printed to the left of the first line. If *line* = 1 each time a line feed is encountered the current pen position is moved at an angle of 90° to the path in an anticlockwise direction. Hence, if *path* = 2 the second line of text is printed below the first line.
- If you do not use a DV command the default text path is horizontal, left to right: line feeds will move the pen position at 90° to the text clockwise. This is equivalent to using the command DV0,0;
- If you omit both parameters the text path is horizontal, left to right.

CP - Relative character movement

CP [*spaces* [, *lines*]] [;]

spaces ; the number of text spaces the pen position is moved.

lines ; the number of text lines the pen position is moved.

- Use this command to re-position the pen. The command moves the pen the specified number of lines and spaces corresponding to the currently selected font's pitch and line spacing. (For proportionally spaced fonts the width of the space character is used to determine the distance moved horizontally). All movement is relative to the current label printing direction.
- *spaces* is the number of spaces that the pen moves. A positive value of *spaces* moves the pen right from the current pen location: a negative value moves it to the left.
- *lines* is the number of lines that the pen moves. A negative value offset moves the pen down from the current pen location: a positive value moves it upwards.
- Values for *spaces* and *lines* are clamped real numbers.
- The command only moves the pen and does not draw lines, irrespective of whether the pen is currently up or down.
- On completion of the command, the previous pen status (up or down) is restored.
- The height of text lines (the vertical distance that a line feed moves the pen position) is fixed for every font. However, it can be adjusted using the ES instruction.
- If you omit all parameters, the pen position moves to the carriage return point and then down one line - hence CP; is equivalent to a carriage return followed by a line feed.

```

10 '-Character Plot -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA1000,5000;";
70 LPRINT "PD;PR2000,0;PU;";
80 LPRINT "CP-15,1;";
90 LPRINT "LBAbove the line"; CHR$(3);
100 LPRINT "CP-14,-2;";
110 LPRINT "LBBelow the line"; CHR$(3);
120 LPRINT CHR$(27); "%0A";
130 LPRINT CHR$(27); "E";
140 END

```

<Sample 55>

CF - Character fill mode**CF** [*fill* [, *pen*]] [;]*fill* : the fill pattern used to fill characters*pen* : the pen used to edge characters

- This command determines how characters are edged and filled.
- Only scalable fonts can be both filled and edged - bitmap fonts and the stick font cannot be edged and can only be filled with raster fill, shading or PCL cross-hatch patterns. Scalable characters can be filled with any pattern specified by the FT command.
- Permissible values of *fill* are 0, 1, 2 and 3. If *fill* = 0 characters are filled with a solid fill and edged. If *fill* = 1 characters are edged but not filled. (Bitmap characters and stick font characters are filled as they cannot be edged.) If *fill* = 2 characters are filled using the current fill type, but not edged. If *fill* = 3 characters are filled, using the current fill type and edged.
- *pen* can be either 0 or 1. A value of 0 specifies the edging pen to be white; a value of 1 selects the black pen for edging. If no pen is specified in the instruction the current pen, as selected with the SP command, is used.
- The width of the edging pen is proportional to the character's point size.
- The fill line width used for cross hatch patterns is set with the PW command.
- If you omit both parameters solid fill and the white edging pen are selected - this is equivalent to CF0.0;
- If the command is not used all characters are solid and unedged.
- A CF command remains in effect until an IN command is used to initialize the printer, a DF command is used to restore the default settings or another CF command is invoked.

```

10 '-Character Fill Mode -
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "SD1,21,2,1,4,140,5,0,6,3,7,52;SS;"
70 LPRINT "PA1000,3000;DT*;FT3,50,45;"
80 LPRINT "CF1,1;LBA*;"
90 LPRINT "PW0.1;CF3,1;LBB*;"
100 LPRINT "PW0.5;LBC*;"
110 LPRINT CHR$(27); "%0A";
120 LPRINT CHR$(27); "E";
130 END

```

<sample 56>

SI - Set absolute character size**SI** [*width*, *height*] [;]*width* : the width in centimeters of printed characters*height* : the height in centimeters of printed characters

- This command allows you to specify the size of the characters that you print with the LB instruction in centimeters.
- *width* specifies character width and is a clamped real number. Negative values of width produce right-to-left mirror image characters.
- *height* specifies the height of a capital letter and is a clamped real number. Negative values of height produce upside-down mirror image characters.
- If you change character size, the line width used to draw stick font characters may also change.
- If you omit the parameters, the character size is as specified by the most recent SD or AD command.
- If an SB1; command has previously been issued, printed characters may not be exactly the size that you specify.
- If an SB1; command has been issued, specifying a negative value will not produce mirror image characters.

```

10 ' -Absolute Character Size-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "PA700,3000;DT#;LBPrinter#"
70 LPRINT "PA700,2000;SI0.75,1;LBPrinter#"
80 LPRINT "SI;SD1,21,2,1,3,12,5,0,6,0,7,52;SS;"
90 LPRINT "PA40003000;LBPrinter#"
100 LPRINT "PA4000,2000;SI1,1.5;LBPrinter#"
110 LPRINT CHR$(27); "%0A";
120 LPRINT CHR$(27); "E";
130 END

<Sample 57>

```

SR - Set relative character size

SR [*width*, *height*] [;]

width; the width of printed characters expressed as a percentage of the x-component of the distance between P1 and P2.

height; the height of printed characters expressed as a percentage of the y-component of the distance between P1 and P2.

- This command allows you to specify the size of the character that you print with the LB instruction relative to the scaling points P1 and P2.
- *width* sets the character width to a percentage of $|P2x - P1x|$ and is a clamped real number. Negative values of width produce right-to-left mirror image characters
- *height* sets the character height to a percentage of $|P2y - P1y|$ and is a clamped real number. Negative values of height produce upside-down mirror image character.
- If you change the positions of P1 and P2, the character size will change unless the scaling points remain in the same relative positions.
- If you move P2 to the left of, or below P1, characters will appear in mirror image.
- If you change the character size, the line width used to draw stick font characters may also change.
- If you omit the parameters, the character size is as specified by the most recent SD or AD command.

```

10 ' -Relative Character Size-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "IP2000,2000,6500,6500;DT@";
70 LPRINT "SR;PA0,2700;LBRELATIVE LABEL SIZE@"
80 LPRINT "IP0,0,5500,5500;PA0,2000;"
90 LPRINT "LBNEW P1 AND P2 CHANGE LABEL SIZE@"
100 LPRINT "PA0,1000;SR2.5;LBNEW SR INSTRUCTION`;CP;"
110 LPRINT CHR$(27); "%0A";
120 LPRINT CHR$(27); "E";
140 END

<Sample 58>

```

SL - Character slant

SL [$\tan \phi$] [;]

$\tan \phi$ - tangent of character slant angle (ϕ)

- This command determines the slant of characters printed with subsequent LB commands.
- ϕ is the angle between characters and the vertical. A positive value of ϕ (and $\tan \phi$) produces forward sloping characters, while a negative value results in characters which slant backwards.
- Values of $\tan \phi$ are clamped real numbers.
- A value of 0 for $\tan \phi$ results in upright characters.
- If you omit the parameter, $\tan \phi$ is take to be 0 and any characters printed will be upright.

```

10 ' -Character Slant-
20 WIDTH "LPT1:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27; "%0B";
50 LPRINT "IN;SP1;";
60 LPRINT "SD1,21,2,1,4,25,5,0,6,0,7,5;SI0.7,1;"
70 LPRINT "PA1000,1000;DT#,1;SL0.36,LBPrinter#"
80 LPRINT "PA1000,300;SL-0.36;LBPrinter#"
90 LPRINT CHR$(27); "%0A";
100 LPRINT CHR$(27); "E";
110 END

<Sample 59>

```

SB - Scalable or bitmap fonts

SB [*n*] [;]

n : font category selection

- This command enables you to restrict the range of selectable fonts to scalable fonts and the stick font only.
- *n* can be 0 or 1. If *n* = 0 you may only select scalable fonts (or the stick font). If *n* = 1 you may select any type of font.
- An SB command can alter the current standard and alternate font selections.
- Setting *n* to 1 allows the selection of bitmapped fonts and has the following side-effects:
 - Bitmap characters cannot be edged.
 - Bitmap characters can only be printed in orientations of 0°, 90°, 180° or 270°.
 - The SL (character slant) command is not applied to bitmap fonts.
 - Bitmap font character sizes are approximate only.
- The DF and IN commands set *n* to 0, - only scalable fonts and the stick fonts may be selected.
- The FI and FN commands implicitly set *n* to 1 if you use either of them to select a bitmap font.
- If you omit *n* from the command, only scalable fonts and the stick font may be selected. This is equivalent to an SB0; command.
- The default setting is scalable fonts and stick font only.

ES - Extra space

ES [*width* [, *height*]] [;]

width : factor by which character spacing is changed.

height : factor by which line spacing is changed

- This command enables you to adjust the space between label characters and lines.
- *width* specifies the factor by which to increase or decrease the character spacing. Positive numbers specify an increase and negative numbers a decrease.
- *height* specifies the factor by which to increase or decrease the line spacing. Positive numbers specify an increase and negative numbers a decrease.
- If you use the command without parameters, no extra space is used when labels are printed. This is equivalent to ES0,0;
- An ES command remains in effect until another ES command is executed, a DF or IN command is executed or the printer is reset, either with the EscE command or from the control panel.

TD - Transparent data

TD [*mode*] [;]

mode : normal or transparent data mode.

- This command enables you to specify whether control characters should perform their normal function or should be printed.

- *mode* is set to either 0 or 1. If *mode* is set to 0, control codes perform their normal functions and are not printed. If *mode*=1, all characters are printed, if possible. No control characters perform any function except the label terminator. Any non-printing or undefined characters appear as a space.
- If you use the command without parameters, control codes will perform their normal functions and will not be printed. This is equivalent to TD0;

```

10 '-Transparent Data-
20 WIDTH "LPT:", 255
30 LPRINT CHR$(27); "E";
40 LPRINT CHR$(27); "%0B";
50 LPRINT "IN;SP1;"
60 '-TD=1 (Transparent) -
70 LPRINT "PA1000,5000;LBMODE : TD1 (Transparent)"; CHR$(13)
80 LPRINT "<Print as characters when labeling>"; CHR$(3)
90 X=1000
100 FOR I = &H0 O &H8
110 LPRINT "SD1,341,2,1,4,10,7,5;PA";X;"",4000;TD1;DT@;LB";CHR$(I);"@";
120 X = X + 150
130 NEXT I
140 LPRINT "TD0;"
150 LPRINT CHR$(27); "%0A";
160 LPRINT CHR$(27); "E";
170 END

```

<Sample 60>

8. INDEX

A

Absolute arc three point	21
absolute commands	8
Absolute direction	45
Anchor corner	30
anchor point.....	7

B

Bezier absolute	24
Bezier relative	24

C

Character fill mode.....	48
Character plot.....	43
Character slant.....	49
Circle plot.....	23
current units.....	7

D

Default set instruction	12
Define alternate font.....	42
Define label terminator.....	44
Define standard font.....	39
Define variable text path	47
Draw absolute arc.....	20
Draw relative arc	21

E

Edge polygon	26
Edge rectangle absolute.....	25
Edge rectangle relative	26
Edge wedge.....	26
Extra space	50

F

fill commands	8
Fill polygon	29
Fill rectangle absolute	27
Fill rectangle relative.....	28
Fill type	31
Fill wedge.....	28

G

graphic units.....	7
graphics window	7, 10, 11

H

hard clip limits	7
------------------------	---

I

image plots.....	11
Initialize command.....	12
Initialize set instruction.....	13
Input relative scaling points.....	14
Input scaling point	14

L

Label origin.....	44
Line attribute.....	32
Line type selection.....	33

M

mnemonic.....	9
---------------	---

P

parameters.....	9
pen	7
Pen down	18
Pen up	18
Pen width	34
Plot absolute.....	19
Polygon mode	25
Polyline encoded.....	22
printer frame	7

R

Raster rill definition	34
Relative arc three point	22
Relative character movement.....	47
relative commands	8
Relative coordinate pen move.....	19
Relative direction	46
Rotate coordinate system	17

S

Scalable or bitmap fonts	50
Scale.....	15
scaling point.....	10
Screened vectors	35
Select alternate font	42
Select pen.....	35

Select primary font..... 43
Select secondary font 43
Select standard font 42
Select unit for pen width 37
separators 9
Set absolute character size..... 48
Set relative character size..... 49
soft clip limits..... 7
Symbol mode 35
syntax 9

T

terminator 9

Transparency mode.....36
Transparent data.....50

U

user units.....7
User-defined line type.....37

W

Window17