5. PR-9104 Features

5.1 Print Feature Control

Print Quality and Font

This printer has a wide variety of print capabilities as show below. The user can select any print mode by combining them.

Quality	Font	Font Style	Pitch	Highlight
Draft		Superscript	10	Double height
NLQ	Courier	Subscript	12	Double width
	Bold PS	Italic	15**	Emphasize
	Prestige		17	Double strike
	Sans Serif*		20**	Underline
			PS	Overline***

* Available only through software command.

** Available in Standard Mode only through software; not available in IBM Mode.

*** Available only in IBM Mode.

This printer has two print quality levels: Draft and NLQ (Near Letter Quality). Which you choose depends on your need. Draft, printed at the fastest speed, is normally used for printing draft documents. NLQ produces the best print quality; it is used to print the final version of formal documents. The printer has four NLQ fonts: Courier, Bold PS, Prestige and Sans Serif and can be selected either by setting the Control Table on the EZ-Set Operator panel or through software. Super/subscript font characters are two-thirds the height of normal characters and are typically used in mathematical expressions, chemical formulae, and footnoting.

Character Pitch

This printer has six character pitches: 10 cpi (Pica), 12 cpi (Elite), 15 cpi (Micron), 17 cpi (Compressed), 20 cpi (Elite Compressed) and Proportional Spacing. The first five pitches are fixed pitch (within a pitch, all characters have the same width). In proportional spacing, character widths vary with the character. An "I", for example, takes up less space than as "M" or a "W". 15 cpi and 20 cpi modes are available only in Standard Mode.

(Print Example)

10 cpi printing (Pica)
12 cpi printing (Elite)
15 cpi printing (Micron)
17 cpi printing (Compressed)
20 cpi printing (Elite Compressed)
Proportional Spacing

Character Highlighting

This printer allows a document to have a variety of print styles through the mixing of fonts and pitches.

Double height printing makes the height of a character twice that of a normal one.

Double width printing makes the width of a character twice that of a normal one.

Double printing uses a double strike with two passes of the print head, feeding the paper 1/216" (0.12 mm) between the first and second pass. (In Standard Mode.)

Emphasized printing is done in one pass of the print head at half speed, which allows horizontally adjacent dots to be printed.

Underline printing produces a continuous line under characters, using the 9th pin of the print head.

Overline printing produces a continuous line over characters using the first pin of the print head. This is available only in the IBM Mode.

(Print Example)

Double Height
Double Width
Double printing
Emphasized Printing
Underline Overline Printing

5.2 Download Characters

If the printer does not contain all of the characters which you need, you can custom design up to 48 (6.0 KB) characters without the buffer option and up to 256 characters (24.0 KB) of both DRAFT and NLQ with the buffer option.

Standard Mode (Epson FX-86e/FX-800)

DRAFT FONT DOWNLOADING:

Download draft font.

Name: ESC+&+0+Cs+Ce+At+P_1+P_2+...+P_{11}
(0 Cs Ce 255)

Dec.: 27, 38, 0, Cs, Ce, At, P_1, P_2,..., P_{11}

Hex.: 1B, 26, 00, Cs, Ce, At, P_1, P_2,..., P_{11}

Input Format: LPRINT CHR\$(27)+"&"+CHR\$(0)+CHR\$(Cs)+CHR\$(Ce)+CHR\$(At)+CHR\$(P_1)+CHR\$(P_2)+...+CHR\$(P_{11})

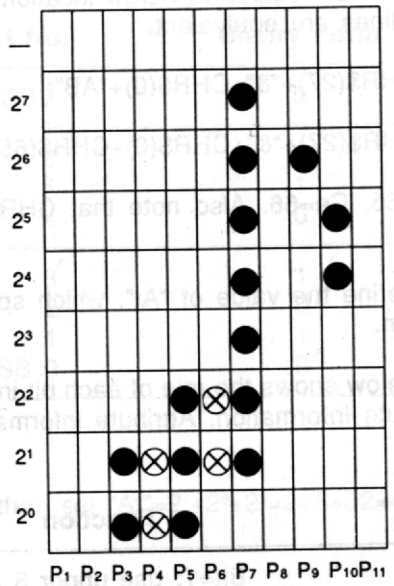
Example:

```
100 REM DEFINITION OF DOWNLOAD CHARACTERS IN DRAFT MODE
101 WIDTH "LPT1:",255
102 OPEN  "LPT1:" AS #1
110 PRINT #1,CHR$(27)+"&"+CHR$(0)+CHR$(65)+CHR$(66);
120 REM STORE IN PLACE OF "A" - ASCII CODE 65
130 PRINT #1,CHR$(170);
140 RESTORE 310
150 FOR I=1 TO 11
160     READ A : PRINT #1,CHR$(A);
170 NEXT I
180 REM STORE IN PLACE OF "B" - ASCII CODE 66
190 PRINT #1,CHR$(42);
200 RESTORE 310
210 FOR I=1 TO 11
220     READ A : PRINT #1,CHR$(A);
230 NEXT I
240 REM SELECT DOWNLOAD CHARACTER
250 PRINT #1,CHR$(27)+"%"+CHR$(1);
260 PRINT #1,"ABABABABABAB"+CHR$(10)+CHR$(10)
270 REM DOUBLE HEIGHT DOUBLE WIDTH PRINT
280 PRINT #1,CHR$(27)+"W1"+CHR$(27)+"w1"
290 PRINT #1,"ABABABABABAB"+CHR$(10);
291 CLOSE
300 END
310 DATA 0,0,3,3,7,6,254,0,64,48,0
```

```
♪♪♪♪♪♪♪♪♪♪♪♪

♪ ♪ ♪ ♪ ♪ ♪ ♪ ♪ ♪ ♪ ♪ ♪
```

Explanation:
To download a character into RAM, you must first design the character. You must quantify each dot column, P_1–P_{11}, by summing the powers of two represented by each dot. Consider the design of the musical note.


—
2^7
2^6
2^5
2^4
2^3
2^2
2^1
2^0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

P_1 =0
P_2 =0
P_3 =3
P_4 =3
P_5 =7
P_6 =6
P_7 =254
P_8 =0
P_9 =64
P_{10} =48
P_{11} =0

Note: See page 5-7, 5th comment.

In our sample program lines 130~170, therefore, use the eleven values P_1–P_{11} (Program line 310 is data of P_1–P_{11}) to define the shape and size of the musical note using the upper 8 pins of the print head. Program lines 190–230 define the same shape and size, but for the lower 8 pins of the print head.

Next you must determine where in RAM the character(s) should be stored. The variables "Cs" and "Ce" are used for this purpose. The value specified for "Cs" indicates the ASCII location into which the starting downloaded character will be stored. The value specified for "Ce" indicates that ASCII location into which the ending downloaded character will be stored. If you are storing a single character, then Cs=Ce.

In our sample program, we created two musical note characters, one using the upper 8 pins of the print head, and the other using the lower 8 pins. These two distinct characters were stored in the ASCII locations where characters "A" and "B" are normally stored (see program line 110). Since "A" resides in ASCII location 65_{DEC} and "B" resides in ASCII location 66_{DEC}, the following program lines are equivalent.

```
110 LPRINT CHR$(27)+"&"+CHR$(0)+"AB";
    or
110 LPRINT CHR$(27)+"&"+CHR$(0)+CHR$(65)+CHR$(66);
```

Notice that Cs=65, Ce=66. Also note that CHR$(65)="A" and CHR$(66)="B".

We must next define the value of "At", which specifies the attribute information.

The illustration below shows the role of each bit in the specification of the attribute information. Attribute information is stored in one byte.

Bit NO.		Function

MSB	7	Bit=1: use upper 8 pins
		Bit=0: use lower 8 pins

	6	Derives print start (Ps) position for proportional spacing, $0 \leqq Ps \leqq 7$
	5	
	4	
	3	Derived print end (Pe) position for proportional spacing, $0 \leqq Pe \leqq 11$
	2	
	1	
LSB	0	$Ps<Pe$

Suppose we wish to create the character (musical note) in proportional spacing mode.

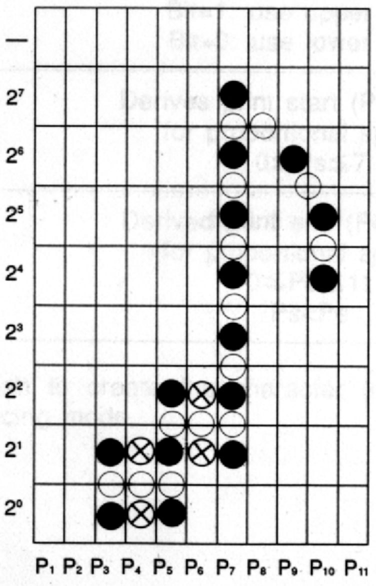
The values for Ps and Pe are derived from n and m (starting and ending print positions respectively) where Ps=n−1 and Pe=m−1. When printing characters from print column P_3 to P_{11} in proportional spacing mode, $Ps=2=(010)_2$ and $Pe=10=(1010)_2$. Thus, the attribute byte is as follows:

Bit No.	Binary Form	Function
MSB 7	0	Use lower 8 pins
6	0	Start of printing in column 3 (3−1)
5	1	
4	0	
3	1	End of printing in column 11 (11−1)
2	0	
1	1	
LSB 0	0	

We then set "At"=$2^1+2^3+2^5=2+8+32=42$.

Comments:

- When you select an NLQ font (COURIER, PRESTIGE, BOLD PS, SANS SERIF*) through the Control Table, the printer will print draft download characters in NLQ mode, see below for detail. (* is available only through software command.)
- The downloadable draft font actually consists of 12 columns of dots, but P_{12} is always set to zero.
- Downloaded draft characters require 12 bytes per character:1 byte for attribute information and 11 bytes for the character design.
- Avoid continuous printing of download characters with high dot densities. 1 line has 8 pins × 11columns × 80 character dots (7040 dots/line). Keep the line density to 25% of maximum (1760 dots/line).
- In single height print mode, avoid using the same pin in two adjacent columns: otherwise, the pin in the second column of the pair will not be fired. In double height print mode, the musical note will print like the figure below. Pins will be fired in all the specified dot positions (● and ⊗) and the printer will add dots (○) automatically (see page 5-16, added dots for detailed information). If you select draft mode, specified dots (⊗) are not printed. However, you have to specify these dots (●) in order to print the character in double height mode or when NLQ is set from the Control Table.


—
2^7
2^6
2^5
2^4
2^3
2^2
2^1
2^0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

NLQ FONT DOWNLOADING:

Defines near letter quality font.

Name: ESC+&+2+Cs+Ce+At+P_{1H}+P_{1L}+...+P_{23L} ($0 \leqq Cs \leqq Ce \leqq 255$)

Dec.: 27, 38, 2, Cs, Ce, At, P_{1H}, P_{1L},..., P_{23L}

Hex.: 1B, 26, 02, Cs, Ce, At, P_{1H}, P_{1L},..., P_{23L}

Input Format: LPRINT CHR$(27)+"&"+CHR$(2)+CHR$(Cs)+
CHR$(Ce)+CHR$(At)+CHR$($P_{1H}$)+CHR$(P_{1L})+...+
CHR$($P_{23L}$)

Example:

```
100 REM DEFINITION OF DOWNLOAD CHARACTERS IN NLQ MODE
101 WIDTH "LPT1:",255
102 OPEN  "LPT1:" AS #1
110 PRINT #1,CHR$(27)+"&"+CHR$(2)+CHR$(65)+CHR$(65);
120 REM STORE IN PLACE OF "A"  - ASCII CODE 65
130 PRINT #1,CHR$(11);
140 RESTORE 260
150 FOR I=1 TO 23*2
160    READ A : PRINT #1,CHR$(A);
170 NEXT I
180 REM SELECT DOWNLOAD CHARACTER
190 PRINT #1,CHR$(27)+"%"+CHR$(1);
200 PRINT #1,CHR$(27)+"x1";
210 PRINT #1,"AAAAAAAAAA"+CHR$(10)+CHR$(10);
220 REM DOUBLE HEIGHT DOUBLE WIDTH PRINT
230 PRINT #1,CHR$(27)+"W1"+CHR$(27)+"w1";
240 PRINT #1,"AAAAAAAAAA"+CHR$(10);
241 CLOSE
250 END
260 DATA 0,12,0,18,0,2,0,2,15,1,16,129,32,129,64,129
270 DATA 0,129,128,129,128,129,128,129,128,129,128,130
280 DATA 65,130,32,242,31,12,32,0,0,0,0,0,0,0,0,0,0,0
```

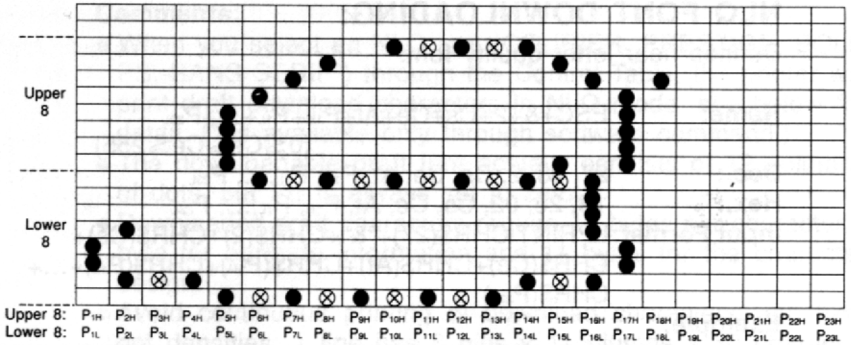
```
9999999999
9999999999
```

Comment:

- This command is operational only when the 32k buffer option (KX-P43) is installed.

Explanation:

An NLQ font downloaded character uses 23 columns and 16 rows of dots. Since a given column contains 16 dots, each column is divided into 2 bytes. For example, column 1 is labeled P_{1H} for the upper 8 dots and P_{1L} for the lower 8 dots. Similarly, column 23 is labeled P_{23H} for the upper 8 dots and P_{23L} for the lower 8 dots. Column 24 is always set to zero; thus, we are working with P_{1H}–P_{23H} and P_{1L}–P_{23L}. Additionally, since each column contains 16 dots which overlap, the printing of an entire character is accomplished with two passes of the print head.


Upper 8
Lower 8
Upper 8: P1H P2H P3H P4H P5H P6H P7H P8H P9H P10H P11H P12H P13H P14H P15H P16H P17H P18H P19H P20H P21H P22H P23H
Lower 8: P1L P2L P3L P4L P5L P6L P7L P8L P9L P10L P11L P12L P13L P14L P15L P16L P17L P18L P19L P20L P21L P22L P23L

Then

P_{1H}	=0	P_{13H}	=128	P_{1L}	=12	P_{13L}	=129
P_{2H}	=0	P_{14H}	=128	P_{2L}	=18	P_{14L}	=130
P_{3H}	=0	P_{15H}	=65	P_{3L}	=2	P_{15L}	=130
P_{4H}	=0	P_{16H}	=32	P_{4L}	=2	P_{16L}	=242
P_{5H}	=15	P_{17H}	=31	P_{5L}	=1	P_{17L}	=12
P_{6H}	=16	P_{18H}	=32	P_{6L}	=129	P_{18L}	=0
P_{7H}	=32	P_{19H}	=0	P_{7L}	=129	P_{19L}	=0
P_{8H}	=64	P_{20H}	=0	P_{8L}	=129	P_{20L}	=0
P_{9H}	=0	P_{21H}	=0	P_{9L}	=129	P_{21L}	=0
P_{10H}	=128	P_{22H}	=0	P_{10L}	=129	P_{22L}	=0
P_{11H}	=128	P_{23H}	=0	P_{11L}	=129	P_{23L}	=0
P_{12H}	=128			P_{12L}	=129		

Program lines 130~170, therefore, use the values P_{1H}–P_{23H} and P_{1L}–P_{23L} to define the shape and size of the character using the lower 8 pins of the print head.

As shown in program line 110, this single character is stored in ASCII location 65, where the character "A" normally resides.

Downloading NLQ font characters requires 1 byte of attribute information which is input as the value of "At". Same as attribute information of draft font, refer to page 5-5 table.

The values Ps and Pe of NLQ attribute information are derived from n and m (starting and ending print positions respectively) using the equation Ps=(n–1)/2 and Pe=(m–1)/2. To ensure that Ps and Pe are integers, the character must be designed so that n and m are odd.

In this example, the g starts in column 1 and ends in column 23, and Ps=0=$(000)_2$ and Pe=11=$(1011)_2$. Thus, the attribute byte is as follows:

Bit No.	Binary Form	Function
MSB 7	0	Use lower 8 pins
6	0	Start of printing in column 1 (1–1)/2
5	0	
4	0	
3	1	End of printing in column 23 (23–1)/2
2	0	
1	1	
LSB 0	1	

We then set "At"=$2^3+2^1+2^0$=8+2+1=11
Refer to program line 130.

Comments:

- Downloaded NLQ font characters require 47 bytes per character: 1 byte for attribute information and 46 bytes for the character design.
- Refer to comments for single height print mode on page 5-7.

ROM CHARACTER GENERATOR SET COPY:

Copies internal ROM CG font into downloadable font area.

Name:	ESC+:+0+0+0
Dec.:	27, 58, 0, 0, 0
Hex.:	1B, 3A, 00, 00, 00
Input Format:	LPRINT CHR$(27)+":"+CHR$(0)+CHR$(0)+CHR$(0)

Example:

```
100 REM CHARACTER GENERATOR SELECTION
101 WIDTH "LPT1:",255
102 OPEN  "LPT1:" AS #1
110 REM ROM CG SET COPY TO DOWNLOAD BUFFER
120 PRINT #1,CHR$(27)+":"+CHR$(0)+CHR$(0)+CHR$(0);
130 REM DOWNLOAD TO "!" IN DRAFT FONT
140 PRINT #1,CHR$(27)+"&"+CHR$(0)+"!!";
150 PRINT #1,CHR$(11);
160 RESTORE 270
170 FOR I=1 TO 11
180     READ A : PRINT #1,CHR$(A);
190 NEXT I
200 REM SELECT DOWNLOAD CG
210 PRINT #1,CHR$(27)+"%"+CHR$(1);
220 PRINT #1,"SELECT DRAFT FONT DOWNLOAD CG !!!"+CHR$(10);
230 REM SELECT ROM CG
240 PRINT #1,CHR$(27)+"%"+CHR$(0);
250 PRINT #1,"SELECT DRAFT FONT ROM CG !!!"+CHR$(10);
251 CLOSE
260 END
270 DATA 0,0,3,3,7,6,254,0,64,48,0
```

```
SELECT DRAFT FONT DOWNLOAD CG ♪♪♪
SELECT DRAFT FONT ROM CG !!!
```

Comments:

- ● All ROM CG font in draft and NLQ modes are copied to the downloadable font area.
- ● Usable capacity of downloadable font does not decrease by using ROM CG set copying.
- ● Upon receipt of the command, all previous downloaded font will be changed to ROM CG font. The usable capacity of downloadable font returns to the initial state.
- ● When altering only part of the ROM CG, use this command before font downloading.

CHARACTER GENERATOR SELECTION:

Selects the character generator.

Name:	ESC+%+n	(n=0 or n=1)
Dec.:	27, 37, n	
Hex.:	1B, 25, n	

Comment:

● "n" specifies the CG mode.
n=0: Resident (internal) CG
n=1: Download CG

IBM Mode

In IBM mode there are many methods of printing down line loaded characters. The various methods are shown in the table below:

Print Method	Selection Command	Dot Columns
Draft Download	ESC+I+4	11 Max.
Draft Download 12 CPI High Speed	ESC+I+5	9 Max.
Text Download fonts	ESC+I+6	11 Max.*
NLQ Download fonts	ESC+I+7	23** (Prints in the same space as draft 11 columns)

* Text download fonts are derived from a Draft download matrix (the printer adds dots to create NLQ characters). Proportional spacing is supported.

** NLQ fonts print as designed (the printer does not add dots). Proportional spacing is not supported.

Draft and Text Character Downloading

Name: ESC+=+n_1+n_2+20+Cs+At_1+At_2+P_1+P_2+...+P_{11} ($0 \leq Cs \leq 255$)

Dec.: 27, 61, n_1, n_2, 20, Cs, At_1, At_2, P_1, P_2,..., P_{11}
Hex.: 1B, 3D, n_1, n_2, 14, Cs, At_1, At_2, P_1, P_2,..., P_{11}

n_1,n_2 indicate the number of bytes of character data to be loaded.

Total count=(total characters×13)+2 and n_2=integer portion of total count/256 and n_1=remainder.

For example, to find n_1 and n_2 for 32 characters:

Total count=(32×13)+2=418

$$\begin{array}{r} 1 \\ 256\overline{)418} \\ 256 \\ \hline 162 \end{array}$$

n_1=162, n_2=1

Note:
- If n_1=n_2=0 all download characters are cleared.
- 20 is a fixed number.

"Cs" indicates the first character position for loading characters. For example to load a character into the position occupied by the ASCII character "A", Cs=65

At_1 is attribute byte 1

Attribute byte 1 specifies:

Bit		
7	0	– Print using upper 8 pins
	1	– Print using lower 8 pins
6-2		Ignored
1,0	00	– Normal character
	01	– Line drawing character. The printer repeats the dots of row 8 in rows 9 to 12 as shown in Figure 1 on page 5-15.
	10	– Shading character. The dots in rows 1 to 4 are repeated in rows 9 to 12 as shown in Figure 2 on page 5-15. NLQ mode is ignored
	11	– Not supported

Note:
- Bit 7 of attribute byte 1 is effective only if bits 0, 1 are 00.

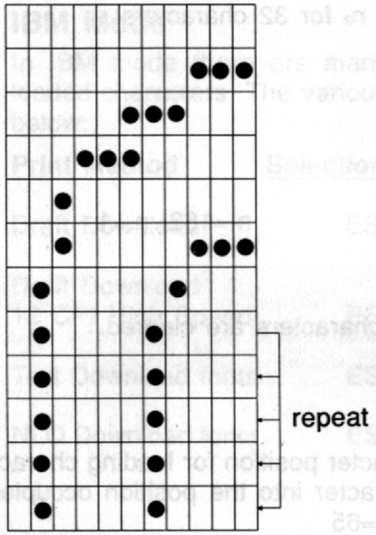


Figure 1

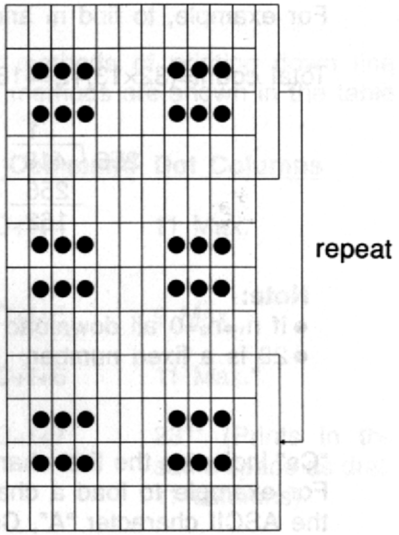


Figure 2

At$_2$ is attribute byte 2

Attribute byte 2 specifies proportional spacing information about the character:

Bit	
7	Reserved
6-4	Specifies the number of column data bytes to ignore (up to 7 leading columns can be ignored).
3-0	Specifies the total number of dot column to be printed. Each character includes a blank column which must be printed. This column is not included by bits 3-0, e.g. For a character 10 columns wide (9 printable+1 blank) bits 3-0 would be 1001. Widths specified greater than 11 are truncated to 11.

Note: To print all 11 columns, bits 6-0 may be set to 0.

Designing Draft and Text download characters.

When designing characters to print in both Draft and Text you should consider **how the printer adds dots to create NLQ characters.** The figures that follow show how the printer treats various situations. Black dots shown must be specified by the font design. Open circles represent dots that are automatically added by the printer in NLQ mode. Note horizontally adjacent dots specified in the font design do not print in draft mode.

Figure 1 illustrate how dots are added to vertical lines and how intersection of vertical and diagonal lines are treated. Figure 2 shows the intersection of two diagonal lines.

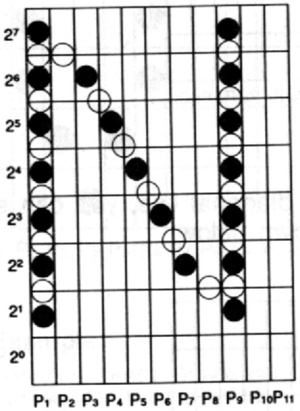

2^7
2^6
2^5
2^4
2^3
2^2
2^1
2^0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

Figure 1

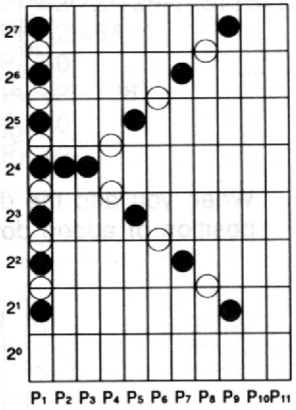

2^7
2^6
2^5
2^4
2^3
2^2
2^1
2^0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

Figure 2

Figure 3 an example of the intersection of vertical and horizontal lines.

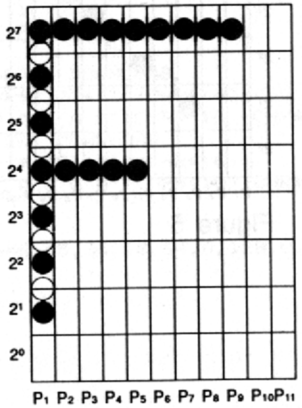

2^7
2^6
2^5
2^4
2^3
2^2
2^1
2^0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

Figure 3

The intersection of diagonal and horizontal lines is printed out as two vertical dots, as shown in Figure 4.

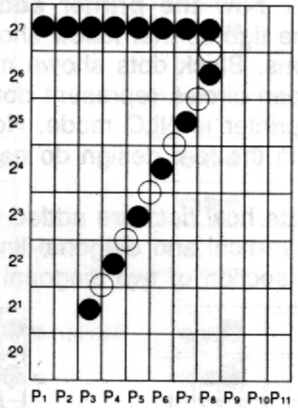

2^7
2^6
2^5
2^4
2^3
2^2
2^1
2^0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

Figure 4

When you add the dots on a diagonal line, you can select the position of added dots as shown below.

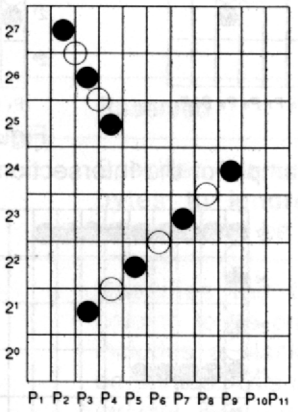
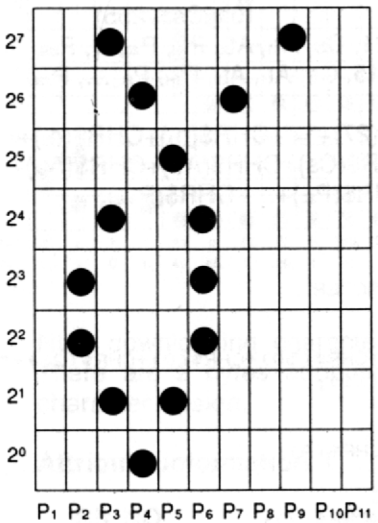

2^7
2^6
2^5
2^4
2^3
2^2
2^1
2^0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

Figure 5

As an example we will provide a basic program that loads and print in draft mode, the greek gamma character shown below.


2^7
2^6
2^5
2^4
2^3
2^2
2^1
2^0
P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8 P_9 P_{10} P_{11}

$P_1 = 0$
$P_2 = 2^2+2^3=4+8=12$
$P_3 = 2^1+2^4+2^7=2+16+128=146$
$P_4 = 2^0+2^6=1+64=65$
$P_5 = 2^1+2^5=2+32=34$
$P_6 = 2^2+2^3+2^4=4+8+16=28$
$P_7 = 2^6=64$
$P_8 = 0$
$P_9 = 2^7=128$
$P_{10} = 0$
$P_{11} = 0$

Example

```
100 REM DRAFT DOWNLOAD CHARACTER
101 WIDTH "LPT1:",255
102 OPEN  "LPT1:" AS #1
110 PRINT #1,CHR$(27)+"="+CHR$(15)+CHR$(0)+CHR$(20)+"A";
120 PRINT #1,CHR$(0)+CHR$(11);
130 RESTORE 240
140 FOR I=1 TO 11
150    READ A : PRINT #1,CHR$(A);
160 NEXT I
170 REM PRINT DRAFT DOWNLOAD
180 PRINT #1,CHR$(27)+"I"+CHR$(4);
190 PRINT #1,"A A A A A"+CHR$(10)+CHR$(13);
200 REM PRINT TEXT DOWNLOAD
210 PRINT #1,CHR$(27)+"I"+CHR$(6);
220 PRINT #1,"A A A A A"+CHR$(10)+CHR$(13);
221 CLOSE
230 END
240 DATA 0,12,146,65,34,28,64,0,128,0,0
```

```
γ γ γ γ γ
γ γ γ γ γ
```

NLQ FONT DOWNLOADING

Defines near letter quality font.

Name: ESC+=+n_1+n_2+21+Cs+At_1+At_2+P_{1H}+P_{1L}+…+P_{23L}
($0 \leqq Cs \leqq 255$)

Dec.: 27, 61, n_1, n_2, 21, Cs, At_1, At_2, P_{1H}, P_{2L},…, P_{23L}
Hex.: 1B, 3D, n_1, n_2, 15, Cs, At_1, At_2, P_{1H}, P_{2L},…, P_{23L}

Input Format: LPRINT CHR$(27)+"="+CHR$(n_1)+CHR$($n_2$)+
CHR$(21)+CHR$(Cs)+CHR$($At_1$)+CHR$(At_2)+
CHR$($P_{1H}$)+CHR$(P_{2L})+…+CHR$($P_{23L}$);

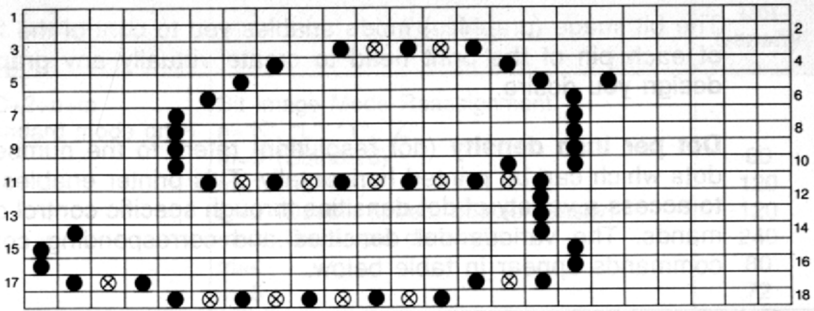
Example

```
100 REM NLQ DOWNLOAD CHARACTER
101 WIDTH "LPT1:",255
102 OPEN  "LPT1:" AS #1
110 PRINT #1,CHR$(27)+"="+CHR$(50)+CHR$(0)+CHR$(21)+"A";
120 PRINT #1,CHR$(0)+CHR$(11);
130 RESTORE 210
140 FOR I=1 TO 23*2
150     READ A : PRINT #1,CHR$(A);
160 NEXT I
170 REM PRINT NLQ DOWNLOAD
180 PRINT #1,CHR$(27)+"I"+CHR$(7);
190 PRINT #1,"A A A A A"+CHR$(10)+CHR$(13);
191 CLOSE
200 END
210 DATA 2,2,1,4,1,0,1,0,48,49,8,65,72,1,8,129
220 DATA 8,1,136,1,136,1,136,1,136,1,137,0,9,144
230 DATA 77,12,50,114,64,0,0,0,0,0,0,0,0,0,0,0
```

9 9 9 9 9

Explanation:
An NLQ font download character uses 23 columns and 18 rows of dots. Column 24 is always blank. P_H specifies the contents of odd rows, P_L the even rows. Of the 18 rows of dots available for each character, only 16 can be printed for normal characters (not line draw or shading). If bit 7 of attribute byte 1 is 0, rows 1~16 will print; if it is set to 1, rows 3~18 will print.

NLQ font Downloading



P_H Data : 2, 1, 1, 1, 48, 8, 72, 8, 8, 136, 136, 136, 136, 137, 9, 77, 50, 64, 0, 0, 0, 0, 0
P_L Data : 2, 4, 0, 0, 49, 65, 1, 129, 1, 1, 1, 1, 1, 0, 144, 12, 114, 0, 0, 0, 0, 0, 0

NLQ downloading characters require 48 bytes per character. There are 2 bytes attribute information and 46 bytes for the character design.

Attribute information 1

Bit NO.	Function
7	0 – Normal character 1 – Descending character
6-2	Ignored
1,1	00 – Normal character print mode (no block graphic character) 01 – Line-drawing character The printer repeats the dots of row 15 in 17, 19, 21 and 23, row 16 in 18, 20, 22 and 24. 11 – Shading character The printer repeats the dots of row 1, 3, 5 and 7 on 17, 19, 21 and 23, row 2, 4, 6 and 8 on 18, 20, 22 and 24. 11 – Not supported

Attribute information 2 is reserved. Always set attribute 2 to zero.

5.3 Bit Image (Graphics)

The bit image (graphics) mode enables you to control the firing of each pin of the print head to create virtually any graphics design you desire.

Dot per inch density (dot resolution) refers to the number of dots which can be printed horizontally. This printer enables you to access a variety of dot densities through specific control commands. The various dot densities and corresponding control commands appear in table below.

Command	Function	Dot Density
ESC+K+n_1+n_2	Standard density designation	60
ESC+L+n_1+n_2	Double density designation	120
ESC+Y+n_1+n_2	Double speed, double density designation	120
ESC+Z+n_1+n_2	Quadruple density designation	240
ESC++*+m+n_1+n_2 (Standard Mode only)	8-Pin Mode Selection:	
	m=0 (Standard)	60
	m=1 (Double)	120
	m=2 (Double speed, double density)	120
	m=3 (Quadruple density)	240
	m=4	80
	m=5	72
	m=6	90
	m=7	144
ESC+^+m+n_1+n_2 (Standard Mode only)	9-Pin Mode Selection:	
	m=0 (Standard)	60
	m=1 (Double)	120
	m=2 (Double speed, double density)	120
	m=3 (Quadruple density)	240
	m=4	80
	m=5	72
	m=6	90
	m=7	144

Dot Resolution (Dots per inch)

Command	Function	Dot Density
ESC+?+n+m (Standard mode only)	Bit Image Mode Reassignment: n="K", "L", "Y", "Z"	
	m=0 (Standard)	60
	m=1 (Double)	120
	m=2 (Double speed, double density)	120
	m=3 (Quadruple density)	240
	m=4	80
	m=5	72
	m=6	90
	m=7	144

Dot Resolution (Dots per inch)

As you can see, each graphics control command uses two bytes, n_1, and n_2, for the designation of the actual number of dots you want printed on a line. The data entered in your program must match this dot specification; if not, in all likelihood your graphics data will contain strange characters.

Determining the values of n_1, and n_2 can be accomplished in the following way. Assume that you want to print N dots on a line, where N is within the proper range for the dot density. Then the outcome of the division below yields the values n_1 and n_2.

$$\begin{array}{r|l} & n_2 \\ \hline 256 & N \\ & \underline{-256 \times n_2} \\ & n_1 \end{array}$$

Where n_2 is the integer quotient and n_1 is the remainder. For those users with a BASIC programming background, n_2=INT (N/256) and n_1=N–(256*n_2).

As an example, suppose we want to print 967 dots per line. Then:

$$\begin{array}{r|l} & 3 \\ \hline 256 & 967 \\ & \underline{768} \\ & 199 \end{array} \quad \text{, so } n_2=3 \text{ and } n_1=199$$

8-Pin Bit Image Mode

Of the 9 pins in the print head, the 8-pin bit image graphics mode uses the upper eight pins only. Each pin corresponds to a power of two. By summing the powers of two corresponding to each of the pins you wish to fire, you will obtain a numerical value which instructs the printer to print one column of dots. Through such techniques in BASIC as looping, numerical values for each column on a line are input and processed. The result is one line of graphics.

Pin No.	Pins	8-Bit Interface	7-Bit Interface
1	•	2^7=128	Not used
2	•	2^6=64	2^6=64
3	•	2^5=32	2^5=32
4	•	2^4=16	2^4=16
5	•	2^3=8	2^3=8
6	•	2^2=4	2^2=4
7	•	2^1=2	2^1=2
8	•	2^0=1	2^0=1
9	•	Not used	Not used

As an example, suppose you want to fire pins 1, 2, 5 and 8 simultaneously. Then you compute the following sum:

Input Code =Pin 1 Code+Pin 2 Code+Pin 5 Code+Pin 8 Code
=$2^7+2^6+2^3+2^0$
=128+64+8+1
=201

Thus, the value 201 is entered in the CHR$ function in order to print a single column of dots resulting from firing pins 1, 2, 5 and 8.

For our final example, refer to the standard density designation in the table on page 5-21. This setting is given by ESC+K+n_1+n_2. Suppose you wish to print 100 columns of dots, where every column fires pins 1 and 8 only.

$$\begin{array}{r} 0 \\ 256\,\overline{)\,100} \\ \underline{0} \\ 100 \end{array}$$, so n_2=0 and n_1=100

You first compute the values of n_1 and n_2.

Our control code ESC+K+n_1+n_2 now translates into:

LPRINT CHR$(27)+"K"+CHR$(100)+CHR$(0);

Next compute the code for firing pins 1 and 8 simultaneously:

Input Code =Pin 1 Code+Pin 8 Code
=2^7+2^0
=128+1
=129

Finally, we incorporate our two calculations into the following program. Note that lines 20 and 30 are necessary for the proper execution of this program on many IBM-compatible computers. Such BASIC statements suppress CR and LF codes and enable printing on a full line without unwanted "breaks". Programs which include statements such as lines 20 and 30 cannot use LPRINTs to print data. In such cases, PRINT# statements must be used. Line 90 is necessary to CLOSE all open files.

```
10 REM STANDARD DENSITY
20 WIDTH "LPT1:",255
30 OPEN "LPT1:" AS #1
40 PRINT #1,CHR$(27)+"K"+CHR$(100)+CHR$(0);
50 FOR I=1 TO 100
60 PRINT #1,CHR$(129);
70 NEXT I
80 PRINT #1,CHR$(10);
90 CLOSE
100 END
```

9-Pin Bit Image Mode (Standard Mode only)

In the 9-pin bit image mode, all 9 pins of the printed head may be fired. The 9 pins in the print head are divided into two portions, the upper 8 pins and the bottom pin.

As in the 8-pin mode, the upper 8 pins correspond to powers of two, ranging from 2^0 to 2^7. The firing of one or more of these 8 pins represents 1 bytes of data. The 9th (bottom-most) pin represents an additional byte of data. When fired, it is represented by the value 2^7. When not fired, it is represented by the value 0. Together, these two bytes determine the dot configuration for a single column of graphics.

Pin No.	Pins	Interface	Byte
1	•	2^7=128	
2	•	2^6=64	
3	•	2^5=32	
4	•	2^4=16	1
5	•	2^3=8	
6	•	2^2=4	
7	•	2^1=2	
8	•	2^0=1	
9	•	2^7=128	
		NOT USED	2

As an example, suppose you want to fire pins 1, 2, 5, 8 and 9 simultaneously. Then you determine the following two values:

Byte 1: Input Code=Pin 1 Code+Pin 2 Code+Pin 5 Code+
(n_1) Pin 8 Code
=$2^7+2^6+2^3+2^0$
=128+64+8+1
=201

Byte 2: Input Code=Pin 9 Code
(n_2) =2^7
=128

Thus, the two bytes for a single column of dots are entered as:
CHR$(201); CHR$(128);

Refer to the 9-pin standard density designation in the table on page 5-21. This setting is given by ESC+^+m+n_1+n_2, where m=0. Suppose you wish to print 100 columns of dots, where every column fires pins 1, 2, 5, 8 and 9 as above.

As in the 8-pin example on page 5-24, n_1=100 and n_2=0. Our control code ESC+^+m+n_1+n_2 now translates into:

LPRINT CHR$(27)+"^"+CHR$(0)+CHR$(100)+CHR(0);

If we incorporate this information into a program, we might have the following:

```
10 REM 9-PIN STANDARD DENSITY
20 WIDTH "LPT1:",255
30 OPEN "LPT1:" AS #1
40 PRINT #1,CHR$(27)+"^"+CHR$(0)+CHR$(100)+CHR$(0);
50 FOR I=1 TO 100
60 PRINT #1,CHR$(201)+CHR$(128);
70 NEXT I
80 PRINT #1,CHR$(10);
90 CLOSE
100 END
```