# CHAPTER 9

# DOT GRAPHICS

**Subjects we'll cover in Chapter 9 include—**

- **This printer's bit image graphics capabilities;**
- **Printing a pre-defined shape;**
- **Plotting a calculated shape;**
- **24-pin dot graphics.**

In Chapter 8 you were introduced to a form of computer graphics; you were able to actually define characters dot by dot. In this chapter you'll learn to use the same principles to make your printer print whole pages of dot graphics! We'll show you how to use dot graphics to create "super download characters". In addition, you'll see how your printer can be used as a graphics plotter. This can have some practical business applications as well as create some terrific computer art!

## COMPARING DOT GRAPHICS WITH DOWNLOAD CHARACTERS

A good understanding of dot graphics requires an understanding of how dot matrix printers work; you may want to review the first few pages in Chapter 8. The principles for dot graphics are the same as those for download characters.

There are some differences in the way they are implemented however. While download commands can be used to define a character between one and fifteen columns of dots wide, dot graphics commands can be used to define a shape as narrow as one column of dots wide or as wide as 3264 dots!

So when do you use graphics and when do you use download characters? Practically anything you can do with graphics you can do with download characters, and vice versa. A clever programmer could actually plot a mathematical curve using download characters or use strings of graphics data as user-defined characters. But why do it the hard way? There are several instances when dot graphics is clearly the best way to approach the problem:

- If the graphics image to be printed is wider than 15 dots.
- If an image is to be printed just one time, as opposed to a frequently used "text" character.

## USING THE DOT GRAPHICS COMMAND

This printer has one command that allows you to use any of the ten graphics modes. The syntax of the command is:

⟨ESC⟩ "*" *n0 n1 n2 m1 m2* .....

Just like many of the other codes you have learned, the command starts with an escape sequence (⟨ESC⟩ "*" in this case). This is followed by *n0,* which specifies the print density as shown in Table 9-1. But unlike the other codes there can be any number of graphics data bytes following the command. That's where *n1* and *n2* come in; they are used to tell the printer how many columns of graphics data to expect.

**Table 9-1**
**Graphics modes**

| *n0* | Pins | Mode | Print density |
|---|---|---|---|
| 0 | 8 | Normal density | 60 dots/inch |
| 1 | 8 | Double density | 120 dots/inch |
| 2 | 8 | Double density | 120 dots/inch |
| 3 | 8 | Quadruple density | 240 dots/inch |
| 4 | 8 | Semi-double density | 80 dots/inch |
| 6 | 8 | CRT graphics | 90 dots/inch |
| 32 | 24 | Normal density | 60 dots/inch |
| 33 | 24 | Double density | 120 dots/inch |
| 38 | 24 | CRT graphics | 90 dots/inch |
| 39 | 24 | Triple density | 180 dots/inch |

■ Specifying the number of columns of dots

To figure the values of *n1* and *n2,* you'll need to figure out how wide your graphics image will be. Then comes the fun part: converting one number (the number of columns of dots) into two! Why is it necessary to use two numbers to tell the printer the number of graphics codes to expect? Because the largest number we can send in one byte (that's what the BASIC CHR$( ) function sends: one byte) is 255. And with the normal density graphics it's possible to have a graphics image as wide as 816 dots. So to figure out how many columns of graphics data to expect, your printer multiplies *n2* by 256 and adds the value of *n1* to the product. If you divide the number of columns by 256, then *n2* is the quotient and *n1* is the remainder (why not let your computer figure it out for you: if the number of columns is assigned to variable X, then *n1* = X MOD 256 and *n2* = INT(X/256)). Table 9-2 might make things even easier.

**Table 9-2**
**Calculating *n1* and *n2***

| If the number of columns, X, ranges from: | Then *n1* is: | and *n2* is: |
|---|---|---|
| 1 to 255 | X | 0 |
| 256 to 511 | X−256 | 1 |
| 512 to 767 | X−512 | 2 |
| 768 to 1023 | X−768 | 3 |
| 1024 to 1279 | X−1024 | 4 |
| 1280 to 1535 | X−1280 | 5 |
| 1536 to 1791 | X−1536 | 6 |
| 1792 to 2047 | X−1792 | 7 |
| 2048 to 2303 | X−2048 | 8 |
| 2304 to 2559 | X−2304 | 9 |
| 2560 to 2815 | X−2560 | 10 |
| 2816 to 3071 | X−2816 | 11 |
| 3072 to 3264 | X−3072 | 12 |

When you are using the 24-pin graphics modes you must send three bytes of data for each dot column. Therefore, you refer to dot columns instead of bytes of graphics data when calculating *n1* and *n2*. (We will explain how these three bytes are interpreted later.)

## ■ Specifying the graphics data

Now that we've told the printer how much data to expect, we better figure out how to send that information! Just as you do with download characters, with dot graphics you have control over firing of every single pin of the print head. When this printer produces 8-pin dot graphics modes, it prints with every third pin. It acts like a printer with nine wires that can only produce 8-pin graphics. In Figure 9-1, you can see that we've labelled each pin on the print head with a number as we did with download characters. And specifying pins to fire is done in the same way: to fire the second pin from the top, for instance, send a CHR$(64). Firing several pins at once is done in a similar fashion. For example, to print the first, third, and fourth dots, add their values (128 + 32 + 16) to send this total: CHR$ (176). This is one byte of graphics data; it would replace *m1* in our format statement.

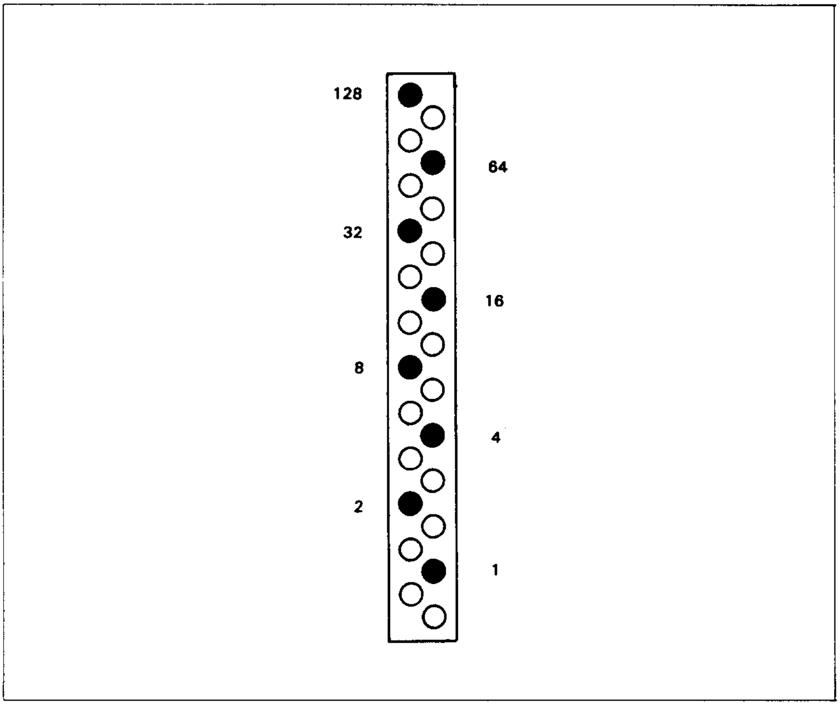
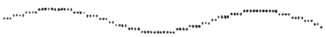


**Figure 9-1.** Starting with the most significant bit at the top, each third pin of the print head is assigned a value which is a power of two in case of 8-pin graphics modes.

A short program should demonstrate how to implement the graphics command. The program below gave us this printout:



```
10 ' Demo of dot graphics
20 PI=3.14159
```

```
30 WID=100
40 OPEN "LPT1:" AS #1 : WIDTH #1,255
50 PRINT#1, CHR$(27);"*";CHR$(0);
60 PRINT#1, CHR$(WID MOD 256);
70 PRINT#1, CHR$(INT(WID/256));
80 FOR I=0 TO WID-1
90 J=1+SIN(I*PI/32)
100 PRINT#1, CHR$(2^INT(J*3.5+.5));
110 NEXT I
120 LPRINT
130 CLOSE#1
```

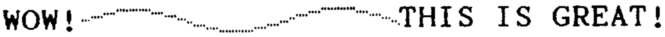
In lines 50 to 70, we've selected 8-pin normal density graphics and said that 100 characters of graphics data would follow. The loop between lines 80 and 110 is repeated to plot 100 points along a curve. This is an example of plotting a very simple mathematical function (a sine wave) to create a design. The mathematical concepts (such as sine and pi) demonstrated here are not important; you don't have to be a math whiz to use this printer's graphics.

■ Combining text and graphics

It's also possible to mix text and graphics in one line. This can be useful for labeling charts or graphs, or even inserting fancy graphics in text. Try adding these lines to our program:

```
45 PRINT#1, "WOW!";
115 PRINT#1, "THIS IS GREAT!";
```

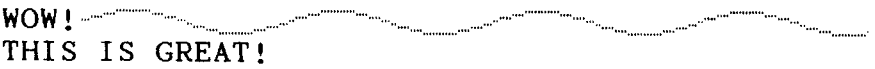
Now if you run the program you should get a printout that looks like this:



```
WOW!                                    THIS IS GREAT!
```

But there is one thing to be careful of: all graphics data must print on the same line. The graphics command is turned off at the end of each line, even if you have specified that more graphics codes follow. To see what we mean, change line 30 to plot 1000 points and run the program.

```
30 WID=1000
```



```
WOW!
THIS IS GREAT!
```

This will make the sine wave pattern long enough to go off the page.

As you can see, your printer printed graphics up to the end of the line, then ignored the rest of the graphics data and returned to normal text on the next line.

## PRINTING A DESIGN OR LOGO

Since you control the firing of every pin, you can print nearly anything with your printer that can draw (and probably better, if you're like most computer users!). You can be used for creating "computer art" or drawing maps. Or, as we'll show you here, you can use dot graphics to print your logo at the top of each letter you print.

Designing an image to print with dot graphics is much like designing download characters. The best way to start is to lay out your image on graph paper. Since you can print eight row of dots with each pass of the print head, draw a heavy horizontal line every eight rows on your graph paper. And it may be helpful to write the dot values (128, 64, 32, etc.) down the left side of each row. Then after you've filled in the "dots" that you want to print, it's time to get out the old calculator again! Just as you did with download characters, add up the values of each column of dots; this makes up one byte.

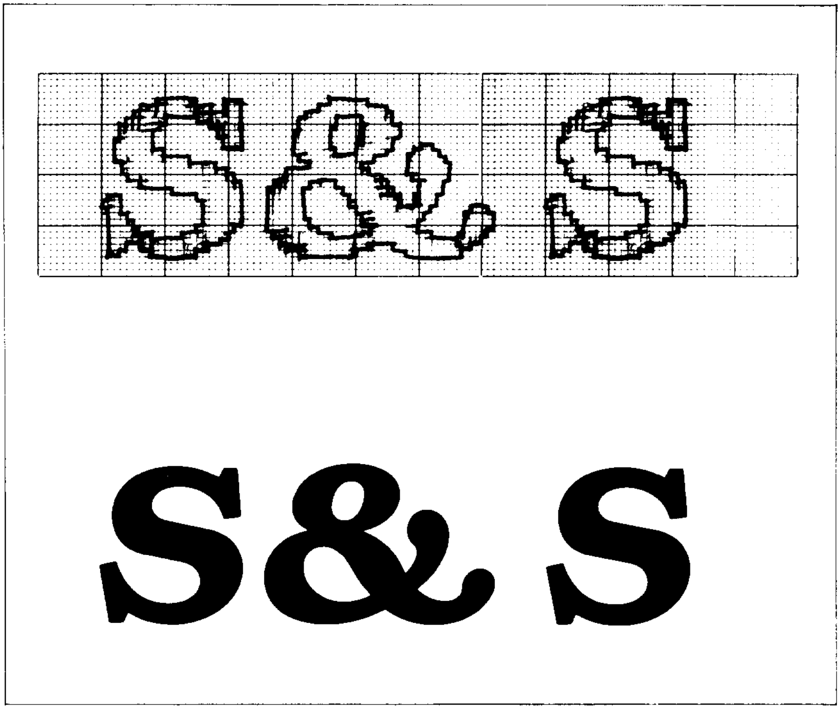


**Figure 9-2.** By laying out the logo on graph paper, you can calculate all of the graphics data.

In the program below, we've taken the logo graphics information and put it into BASIC DATA statements. The program itself is short and simple. The loop starting at line 100 reads the data statements into a string array variable called LOGO$. In line 170 we change the line spacing to 24/180 inch so that the lines of graphics data will connect vertically. The actual printing is done in the loop between lines 180 and 210; line 190 sends the graphics control code to the printer and line 200 sends one line of graphics data.

The printout from the program is shown right below the program.

```
10 ' Prints S&S logo
20 LINE.8$=CHR$(27)+"3"+CHR$(24)"
30 ' Set line spacing to 1/6 inch
40 LINE.12$=CHR$(27)+"A"+CHR$(12)+CHR$(27)+"2"
50 ' Select dot graphics
60 GRAPHIC$=CHR$(27)+CHR$(42)+CHR$(0)
70 DIM LOGO$(4)
80 WIDTH "LPT1:",255
```

```
90 ' Read data
100 FOR ROW=1 TO 4
110 FOR COLUMN=1 TO 100
120 READ P
130 LOGO$(ROW)=LOGO$(ROW)+CHR$(P)
140 NEXT COLUMN
150 NEXT ROW
160 ' Print row
170 LPRINT LINE.8$;
180 FOR ROW=1 TO 4
190 LPRINT GRAPHIC$;CHR$(100);CHR$(0);
200 LPRINT LOGO$(ROW)
210 NEXT ROW
220 LPRINT LINE.12$
230 ' Row 1
240 DATA   0,  0,  0,  0,  1,  3,  7,  7,  7, 15
250 DATA  14, 14, 14, 14, 14,  7,  7,  3,  3, 15
260 DATA  15, 15,  0,  0,  0,  0,  0,  0,  0,  0
270 DATA   0,  1,  3,  3,  7,  7, 15, 14, 14, 14
280 DATA  14, 15,  7,  7,  7,  3,  0,  0,  0,  0
290 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0,  0
300 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0,  0
310 DATA   0,  0,  0,  0,  1,  3,  7,  7,  7, 15
320 DATA  14, 14, 14, 14, 14,  7,  7,  3,  3, 15
330 DATA  15, 15,  0,  0,  0,  0,  0,  0,  0,  0
340 ' Row 2
350 DATA   0,  0, 60,255,255,255,255,255,143, 15
360 DATA   7,  7,  7,  7,  3,  3,  3,131,193,241
370 DATA 240,240,  0,  0,  0,  0,  0,  0,  0,  1
380 DATA 121,253,253,255,255,255,143,  7,  7,  7
390 DATA  31,253,252,248,248,240,192,  0,  7, 15
400 DATA  31, 31, 15,  7,  3,  0,  0,  0,  0,  0
410 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0,  0
420 DATA   0,  0, 60,255,255,255,255,255,143, 15
430 DATA   7,  7,  7,  7,  3,  3,  3,131,193,241
440 DATA 240,240,  0,  0,  0,  0,  0,  0,  0,  0
450 ' Row 3
460 DATA   0, 31, 31,  3,129,128,192,192,192,192
470 DATA 192,224,224,224,224,240,255,255,255,255
480 DATA 255,127,  0,  0,  0,  0, 63,127,255,255
490 DATA 255,255,193,128,128,128,128,192,224,240
500 DATA 252,255,255,255,127, 63, 31,  7,  7, 31
510 DATA 254,252,248,224,128,  0,  0,  3,  7,  7
520 DATA   7,  3,  0,  0,  0,  0,  0,  0,  0,  0
530 DATA   0, 31, 31,  3,129,128,192,192,192,192
540 DATA 192,224,224,224,224,240,255,255,255,255
```

```
550 DATA 255,127,  0,  0,  0,  0,  0,  0,  0,  0
560 ' Row 4
560 DATA   0,248,248,240,224,224,112,112, 56, 56
570 DATA  56, 56, 56,120,120,240,240,224,224,192
580 DATA 128,  0,  0,  0,  0,  0,192,224,240,240
590 DATA 240,248,248,248,120,120, 56, 56, 56, 56
600 DATA  48,112,224,224,224,224,240,240,248,248
610 DATA 120,120, 56, 56, 56, 56,120,240,224,224
620 DATA 192,128,  0,  0,  0,  0,  0,  0,  0,  0
630 DATA   0,248,248,240,224,224,112,112, 56, 56
640 DATA  56, 56, 56,120,120,240,240,224,224,192
650 DATA 128,  0,  0,  0,  0,  0,  0,  0,  0,  0
```

## PLOTTING WITH YOUR PRINTER

This section of the manual gets into more serious BASIC programming just because it's required in order to have the computer act as a plotter driver. Don't be intimidated; while it's beyond the scope of this manual to teach BASIC, if you try the examples and take it slowly you should be doing some fancy plotting of your own before you know it.

If designing and calculating dot graphics images by laying them out on graph paper seems too tedious to you, then let the computer do the work for you! With your computer doing the calculations and your printer plotting the output, you can come up with some terrific business graphs, charts, and mathematical function plots.

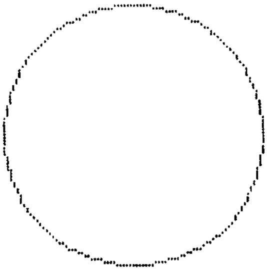
The best way to do this is to set up an array in memory. This is your "graph paper." The first thing to do is to determine how big you want your output to be; this will determine the size of your array. (If you have grandiose plans to fill an entire page with plotter output, you better have lots of memory in your computer. With 60 dots per inch horizontally and 60 dots per inch vertically, it takes at least 480 bytes of memory for each square inch of plotted area. That doesn't sound so bad — but an area 12 inches square requires over 64K!)

Your array should be two-dimensional (just like graph paper) where one dimension will be the number of columns of dots and the other dimension is the number of printing lines (remember that you can have up to eight rows of dots per printed line with the 8-pin graphics mode).

Here's a program that will use calculated-shape graphics to plot a circle. As you'll see, by changing a few lines it can be used to plot virtually any shape.

```
10 ' Plotting program
20 ' Set program constants
30 MAXCOL%=90            :MAXROW%=14
40 DIM BIT%(MAXCOL%,MAXROW%)
50 MASK%(1)=128          :MASK%(5)=8
60 MASK%(2)=64           :MASK%(6)=4
70 MASK%(3)=32           :MASK%(7)=2
80 MASK%(4)=16           :MASK%(8)=1
90 LX=20                 :LY=20
100 LXFAC=90/LX           :LYFAC=90/LY
110 ' Plot curve
120 GOSUB 600
130 ' Send bit image map to printer
140 OPEN "LPT1:" AS #1 : WIDTH #1,255
150 PRINT#1, CHR$(27);"3";CHR$(24);
160 FOR ROW%=0 TO MAXROW%
170 PRINT#1, CHR$(27);"*";CHR$(0);
  CHR$(MAXCOL%);CHR$(0);
180 FOR COL%=1 TO MAXCOL%
190 PRINT#1, CHR$(BIT%(COL%,ROW%));
200 NEXT COL%
210 PRINT#1, CHR$(10)
220 NEXT ROW%
230 PRINT#1, CHR$(27);"A";CHR$(12);CHR$(27);"2"
240 CLOSE#1 : END
250 '
260 ' Subroutine to draw a line from X1,Y1 to
       X2,Y2
270 '
280 XL=X2-X1              :YL=Y2-Y1
290 NX=ABS(XL*LXFAC)      :NY=ABS(YL*LYFAC)
300 IF NX<NY THEN NX=NY
310 NS%=INT(NX+1)
320 DX=XL/NS%             :DY=YL/NS%
330 FOR I%=1 TO NS%
340 X1=X1+DX              :Y1=Y1+DY
350 GOSUB 400
360 NEXT I%
```

```
370 RETURN
400 '
410 ' Subroutine to plot a point at X1,Y1
420 '
430 XX=X1*LXFAC        :YY=Y1*LYFAC
440 COL%=INT(XX)+1
450 ROW%=INT(YY/8)
460 XIT%=INT(YY-ROW%*8)+1
470 BIT%(COL%,ROW%)=BIT%(COL%,ROW%)
  OR MASK%(XIT%)
470 RETURN
600 '
610 ' Subroutine to plot a circle
620 '
630 RAD=9
640 X1=19              :Y1=10
650 FOR ANG%=0 TO 360 STEP 10
660 RANG=ANG%*6.28/360
670 X2=RAD*COS(RANG)+10 :Y2=RAD*SIN(RANG)+10
680 GOSUB 250
690 NEXT ANG%
700 RETURN
```



■ How the program works

In the program above, we've created an array called BIT%, which is dimensioned in line 40. You'll note that instead of using numeric constants to dimension the array, we used the variable MAXCOL% and MAXROW%. This way, if your computer has enough memory and you want to plot a larger image, all you need to change are the values in line 30. The array MASK% contains the values of the dots. In lines 90 and 100 we've defined some other variables you'll be interested in: LX, LXFAC, LY, and LYFAC are used as scaling factors. By changing these values, you can change the size of your printed image or even distort it (you can, for example, make our circle print as an ellipse). Experiment a little bit!

The main calculations for plotting the image are done in the subroutine starting at program line 600. This is where you put the formulas that you want to plot. By changing just the lines after 600 (with some creative mathematics!) you can plot any function — limited only by your imagination. Some examples are shown at the end of this section.

What the program section starting at line 600 actually does is to calculate starting and ending points for a line (in our circle the "lines" are very short — sometimes the starting and ending points are the same). The coordinates of the starting point of the line are assigned to variables X1 and Y1. The line ends at point X2, Y2. When these coordinates have been calculated, a subroutine call is made to line 250. This subroutine calculates the coordinates of individual points along that line.

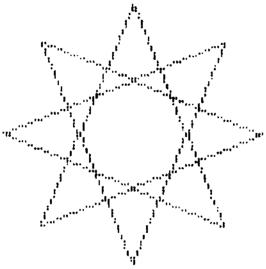
After these coordinates have been determined, the subroutine at line 400 is called. This routine turns "on" an individual dot in our array called BIT%. (Keep in mind that no printing has been done yet; the computer is still drawing the image on its "graph paper" in memory.) The way an individual dot is turned on is using the logical OR function in line 470.

When all the points have been plotted in memory, printing begins at line 130. We first set the line spacing to 24/180 inch using the ⟨ESC⟩ "3" command. This is so that there are no gaps between rows of dots. Then the loop from line 150 to line 220 prints the dot graphics image one line at a time.

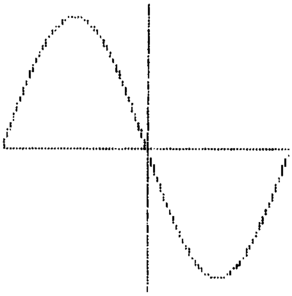
As you can see, by taking the program in small pieces and analyzing it, programming does not have to be difficult. If you want to try some other plots, try these (replace lines after 600 with the lines below). The printouts from each program are shown below the listing.

```
600 '
610 ' Subroutine to plot a star
620 '
630 RAD=9
640 FOR ANG%=0 TO 360 STEP 45
690 RANG=ANG%*3.14159/180
700 RANG2=(ANG%+135)*3.14159/180
710 X1=RAD*COS(RANG)+10
720 Y1=RAD*SIN(RANG)+10
730 X2=RAD*COS(RANG2)+10
```

```
740 Y2=RAD*SIN(RANG2)+10
750 GOSUB 250
760 NEXT ANG%
770 RETURN
```



```
600 '
610 ' Subroutine to plot a sine wave
620 '
630 X1=0 :Y1=10 :X2=20 :Y2=10
640 GOSUB 250
650 X1=10 :Y1=0 :X2=10 :Y2=20
660 GOSUB 250
670 X1=0 :Y1=10
680 FOR X2=0 TO 20 STEP .2
690 Y2=10-9*SIN(3.14159*X2/10)
700 GOSUB 250
710 NEXT X2
720 RETURN
```



## USING THE 24-PIN GRAPHICS MODE

Up until now all of the dot graphics printing we have done has been with the 8-pin graphics modes. This can give you some pretty sharp images at great speed. Sometimes though, you may want to create an image with even higher resolution. This printer has 24-pin graphics modes you can use, as shown in Table 9-1.

The 24-pin dot graphics mode uses all 24 pins in the printhead. The 24 pins are mapped as three eight-bit bytes stacked vertically (as shown in Figure 9-3). This means that for each pin column of 24-pin graphics (as specified by *n1* and *n2*) you must send three bytes of graphics data.

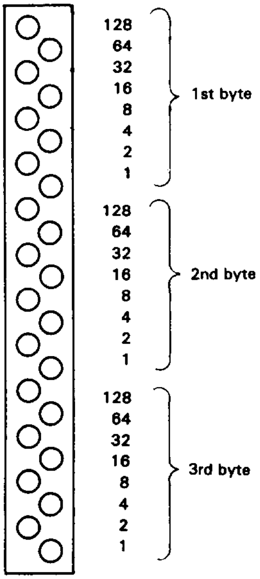


**Figure 9-3**. With the 24-pin graphics modes, each pin column of graphics is divided into three bytes.

Now, let's try to print the 24-pin graphics. Here is the program to print the logo with 24-pin dot graphics.

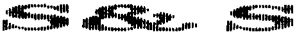
```
10 ' Prints S&S logo (24-pin)
20 LINE.8$=CHR$(27)+"3"+CHR$(24)"
30 ' Set line spacing to 1/6 inch
40 LINE.12$=CHR$(27)+"A"+CHR$(12)+CHR$(27)+"2"
50 ' Select dot graphics
60 GRAPHIC$=CHR$(27)+CHR$(42)+CHR$(32)
70 DIM LOGO1$(2) : DIM LOGO2$(2)
80 WIDTH "LPT1:",255
90 ' Read data
100 FOR ROW=1 TO 2
110 FOR COLUMN=1 TO 150
120 READ P
130 LOGO1$(ROW)=LOGO1$(ROW)+CHR$(P)
140 NEXT COLUMN
150 FOR COLUMN=151 TO 300
160 READ P
170 LOGO2$(ROW)=LOGO2$(ROW)+CHR$(P)
180 NEXT COLUMN
190 NEXT ROW
200 ' Print row
210 LPRINT LINE.8$
220 FOR ROW=1 TO 2
230 LPRINT GRAPIHC$;CHR$(100);CHR$(0);
240 LPRINT LOGO1$;LOGO2$
250 NEXT ROW
260 LPRINT LINE.12$
270 END
280 ' Row 1
290 DATA   0,  0,  0,   0,  0, 31,   0, 60, 31
300 DATA   0,255,  3,   1,255,129,   3,255,128
310 DATA   7,255,192,   7,255,192,   7,143,192
320 DATA  15, 15,192,  14,  7,192,  14,  7,224
330 DATA  14,  7,224,  14,  7,224,  14,  3,224
340 DATA   7,  3,240,   7,  3,255,   3,131,255
350 DATA   3,193,255,  15,241,255,  15,240,255
360 DATA  15,240,127,   0,  0,  0,   0,  0,  0
370 DATA   0,  0,  0,   0,  0,  0,   0,  0, 63
380 DATA   0,  0,127,   0,  0,255,   0,  1,255
390 DATA   0,121,255,   1,253,255,   3,253,193
400 DATA   3,255,128,   7,255,128,   7,255,128
410 DATA  15,143,128,  14,  7,192,  14,  7,224
420 DATA  14,  7,240,  14, 31,252,  15,253,255
430 DATA   7,252,255,   7,248,255,   7,248,127
440 DATA   3,240, 63,   0,192, 31,   0,  0,  7
450 DATA   0,  7,  7,   0, 15, 31,   0, 31,254
460 DATA   0, 31,252,   0, 15,248,   0,  7,224
```

```
470 DATA   0,  3,128,  0,  0,  0,  0,  0,  0
480 DATA   0,  0,  3,  0,  0,  7,  0,  0,  7
490 DATA   0,  0,  7,  0,  0,  3,  0,  0,  0
500 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
510 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
520 DATA   0,  0,  0,  0,  0,  0,  0,  0, 31
530 DATA   0, 60, 31,  0,255,  3,  1,255,129
540 DATA   3,255,128,  7,255,192,  7,255,192
550 DATA   7,143,192, 15, 15,192, 14,  7,192
560 DATA  14,  7,224, 14,  7,224, 14,  7,224
570 DATA  14,  3,224,  7,  3,240,  7,  3,255
580 DATA   3,131,255,  3,193,255, 15,241,255
590 DATA  15,240,255, 15,240,127,  0,  0,  0
600 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
610 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
620 DATA   0,  0,  0
630 ' Row 2
640 DATA   0,  0,  0,248,  0,  0,248,  0,  0
650 DATA 240,  0,  0,224,  0,  0,224,  0,  0
660 DATA 112,  0,  0,112,  0,  0, 56,  0,  0
670 DATA  56,  0,  0, 56,  0,  0, 56,  0,  0
680 DATA  56,  0,  0,120,  0,  0,120,  0,  0
690 DATA 240,  0,  0,240,  0,  0,224,  0,  0
700 DATA 224,  0,  0,192,  0,  0,128,  0,  0
710 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
720 DATA   0,  0,  0,  0,  0,  0,192,  0,  0
730 DATA 224,  0,  0,240,  0,  0,240,  0,  0
740 DATA 240,  0,  0,248,  0,  0,248,  0,  0
750 DATA 248,  0,  0,120,  0,  0,120,  0,  0
760 DATA  56,  0,  0, 56,  0,  0, 56,  0,  0
770 DATA  56,  0,  0, 48,  0,  0,112,  0,  0
780 DATA 224,  0,  0,224,  0,  0,224,  0,  0
790 DATA 224,  0,  0,240,  0,  0,240,  0,  0
800 DATA 248,  0,  0,248,  0,  0,120,  0,  0
810 DATA 120,  0,  0, 56,  0,  0, 56,  0,  0
820 DATA  56,  0,  0, 56,  0,  0,120,  0,  0
830 DATA 240,  0,  0,224,  0,  0,224,  0,  0
840 DATA 192,  0,  0,128,  0,  0,  0,  0,  0
850 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
860 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
870 DATA   0,  0,  0,  0,  0,  0,248,  0,  0
880 DATA 248,  0,  0,240,  0,  0,224,  0,  0
890 DATA 224,  0,  0,112,  0,  0,112,  0,  0
900 DATA  56,  0,  0, 56,  0,  0, 56,  0,  0
910 DATA  56,  0,  0, 56,  0,  0,120,  0,  0
920 DATA 120,  0,  0,240,  0,  0,240,  0,  0
```

```
930 DATA 224,  0,  0,224,  0,  0,192,  0,  0
940 DATA 128,  0,  0,  0,  0,  0,  0,  0,  0
950 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
960 DATA   0,  0,  0,  0,  0,  0,  0,  0,  0
970 DATA   0,  0,  0
```

This program is similar to the previous one. In this program, we've taken the logo graphics information and put it into BASIC DATA statements. The program itself is short and simple. The loop starting at line 100 reads the data statements into string arrays variable called LOGO1$ and LOGO2$. In line 210 we change the line spacing to 24/180 inch so that the lines of graphics data will connect vertically. The actual printing is done in the loop between 210 and 250; line 230 sends the graphics control code (24-pin normal density) to the printer and line 240 sends one line of graphics data.

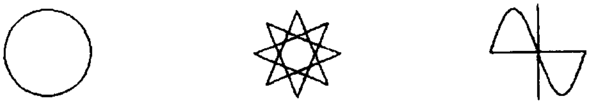
The printout from this program is shown below.



Now, let's use your printer to plot with the 24-pin graphics mode. Load the plotting program again, and change the lines 40, 160, 170, and 190, then add the following two lines to the previous plotting program. You can get the high-resolution results as shown below!

```
40 DIM BIT%(MAXCOL%*3,MAXROW%)
160 FOR ROW%=0 TO MAXROW%-1 STEP 3
170 PRINT#1, CHR$(27);"*";CHR$(39);CHR$(MAXCOL%);CHR$(0);
190 PRINT#1, CHR$(BIT%(COL%,ROW%+BYTE%));
```

```
185 FOR BYTE%=0 TO 2
195 NEXT BYTE%
```



## COMPATIBILITY WITH EXISTING SOFTWARE

With its ability to print ten different graphics densities, this printer's graphics abilities are advanced indeed. There are many programs, in fact, that are unable to use this printer's single graphics command ⟨ESC⟩ "*" for selecting the proper density. To maintain compatibility with this software, there are individual commands to select each of this printer's common graphics densities. These commands, which are shown in Table 9-3, can be used interchangeably with the corresponding ⟨ESC⟩ "*" command. Like the commands you are already familiar with, these new commands are followed by two bytes to specify the number of graphics data bytes to print and then the data.

**Table 9-3**
**Alternative graphics commands**

| Density | Single command | Individual command |
|---|---|---|
| Noraml | ⟨ESC⟩"*" CHR$(0) *n1 n2 m1 m2* . . . | ⟨ESC⟩"K" *n1 n2 m1 m2* . . . |
| Double | ⟨ESC⟩"*" CHR$(1) *n1 n2 m1 m2* . . . | ⟨ESC⟩"L" *n1 n2 m1 m2* . . . |
| Double | ⟨ESC⟩"*" CHR$(2) *n1 n2 m1 m2* . . . | ⟨ESC⟩"Y" *n1 n2 m1 m2* . . . |
| Quadruple | ⟨ESC⟩"*" CHR$(3) *n1 n2 m1 m2* . . . | ⟨ESC⟩"Z" *n1 n2 m1 m2* . . . |

## REDEFINING ALTERNATE GRAPHICS CODES

At the end of this chapter, we'll discuss one mode that the printer offers to help you solve potential graphics problems. A redefining code allows you to change the density for graphics programs that use one of the four alternate codes. The command looks like this:

⟨ESC⟩ "?" *n0 n1*

Where *n0* is one of the four letters, "K", "L", "Y", or "Z", and *n1* is one of the numbers used with the ⟨ESC⟩ "*" command, 0 to 4, 6, 32, 33, 38, and 39.

This is a quick way to change the aspect ratio of the design that you are printing. Changing the graphics mode will change the width without changing the height. However, you should make this change with caution.

If you change one of the 8-pin graphics mode to a 24-pin graphics mode, without changing the program that supplies the graphics data, you will print garbage (if the program prints at all). Remember, the 24-pin graphics modes require three times as much graphics data as the 8-pin graphics modes, and also, the data is arranged differently.

# MEMO

# CHAPTER 10

# CARING FOR YOUR PRINTER

**Subjects we'll cover in Chapter 10 include—**

- **Cleaning the printer;**
- **Changing the ribbon;**
- **Replacing the print head.**

As any good mechanic will tell you, dust and heat are the biggest enemies of any mechanism. And your printer is no exception. The best maintenance is *preventive maintenance*, so the first step in keeping your printer healthy and happy is to make sure it's in a clean, dust-free location. The range of temperature should be comfortable for both you and your computer/printer system. (Please refer to Chapter 1 for more tips on locating your printer.)

## CLEANING THE PRINTER

Another important rule for keeping your printer young and healthy is to clean it regularly—inside and out. Just use a damp towel every week or so (you can moisten the towel with alcohol for stubborn dirt, *but* be careful not to get any alcohol on the printer mechanism).

Use a soft brush to remove dust and lint from inside the printer, but be very careful not to bend or injure any electronic parts or wiring. It doesn't take much to do expensive damage, so don't fuss where you're not supposed to—besides periodic cleaning, the only other maintenance you'll have to do will be changing the ribbon cartridge, or the print head.

## REPLACING THE RIBBON

This printer uses an "endless" ribbon cartridge, which means that the inked ribbon inside is recycled automatically. Eventually, though, printing will become too faint to read easily and you'll want to change the ribbon.

However, it is less expensive and more economical to replace only the ribbon portion inside the catridge. If this money saving method does interest you, read on.

Follow this procedure to remove the old ribbon and insert the new one in the original cartridge (not recommended for people with ten thumbs!).

1. First, obtain from your dealer the correct type of ribbon "sub-cassette"(*not* the spool-type ribbons used with some other printers).
2. Grasp both ends of the ribbon cartridge and pull the cartridge up and out of the printer. (Refer to Chapter 1 for illustrations of installing the refilled ribbon cartridge.)
3. Unhook the seven tabs of the cartridge cover carefully, and pry open the cartridge cover. Figure 10-1 shows the numerous tabs for unhooking.

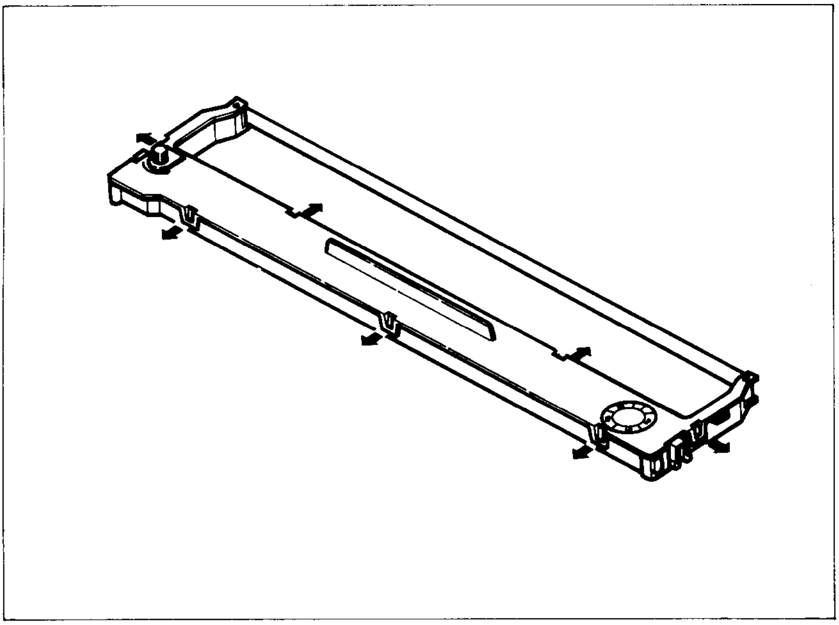


**Figure 10-1.** Unhook tabs to pry open the cartridge.

4. Press hard against the end of the idler gear holder to make a space between the holder and the ribbon drive gear, and remove the old ribbon and holder. See Figure 10-2.

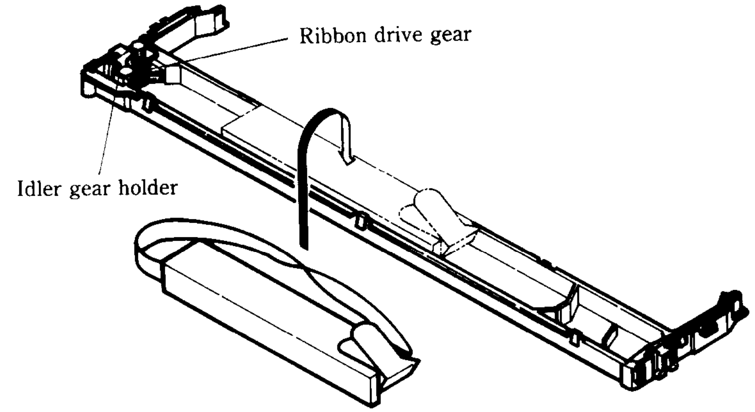


**Figure 10-2.** Replace the ribbon sub-cassette.

5. Clean the inside of the cartridge, the area around the cartridge, and the ribbon drive gear and vicinity.
6. Take the new ribbon and holder out of the wrapper, remove the adhesive tape on the joint, and place the ribbon holder into the cassette as shown in Figure 10-2.
7. Pull out the ribbon and thread it as shown in Figure 10-3. It's easy for the ribbon to get twisted along its path, so be careful.
8. Firmly pull the idler gear towards you and guide the ribbon between the idler gear and the ribbon drive gear.
9. Remove the top and the bottom of the ribbon holder.
10. Replace the ribbon cartridge top cover.

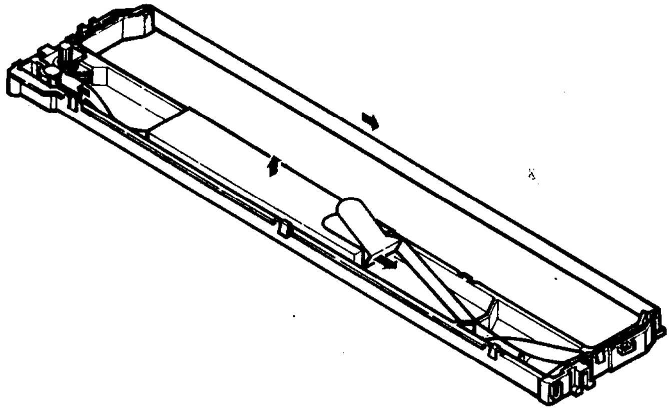


**Figure 10-3.** Make sure that the ribbon is not twisted when you thread it through its path.

11. Now you're almost finished! Mark the next larger number on the silver label at the right-hand side of the cartridge cover to indicate the number of times the ribbon has been replaced. Five replacements is the maximum, after which you should buy a completely new cartridge.

## REPLACING THE PRINT HEAD

The dot matrix print head has a remarkably long life. It will print perhaps 200,000,000 dots per wire before you have to replace it. You'll know when that time comes when printing is too faint even after you replace the ribbon cartridge.

***Warning:*** The print head gets hot during operation, so let it cool off before you touch it.

To replace the print head, start by turning off the Power switch and unplugging the power cord. Then follow this procedure:

1. Remove the printer cover and the ribbon cartridge.
2. Remove the two screws fastening the print head.

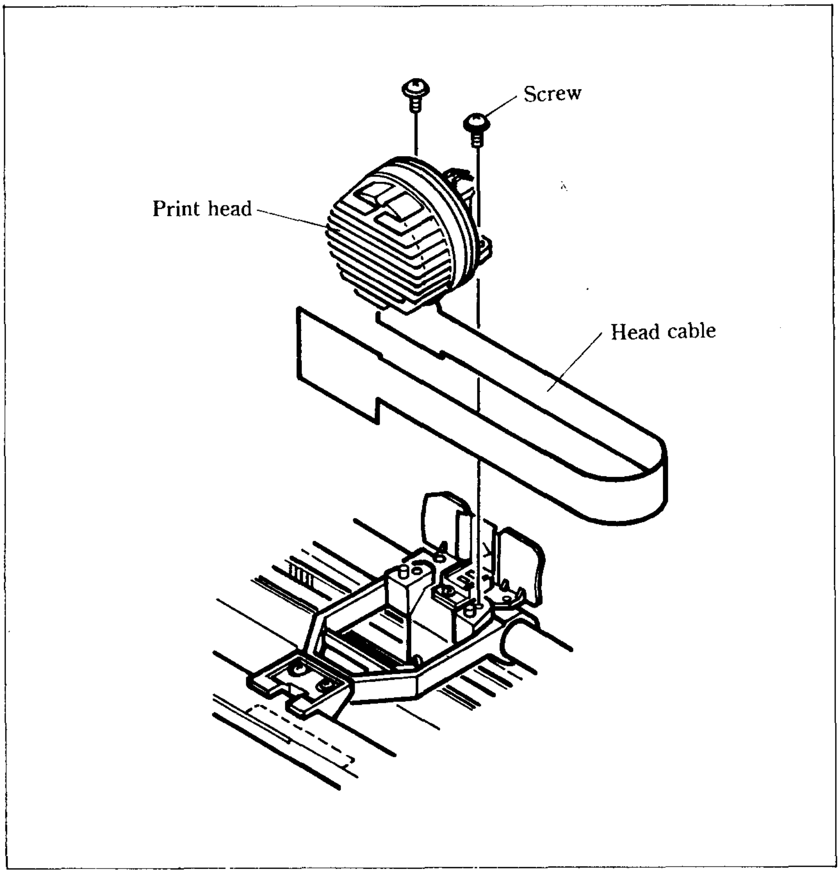


**Figure 10-4.** Replacement of the print head is simple.

3. Holding the print head and the head cable board securely, unplug the head cable.
4. Connect the cable of a new print head to the head cable board and fasten it reversing the above procedures.

Be absolutely sure that the connection between the print head and the cable is secure. A loose cable will cause you problems later.

# MEMO

# APPENDIX A

# DIP SWITCH SETTINGS

The DIP (dual in-line package) switches control some of the functions of the printer. A DIP switch contains several individual switches. This printer has two DIP switches with 8 individual switches. Figure A-1 is a drawing of a typical DIP switch.

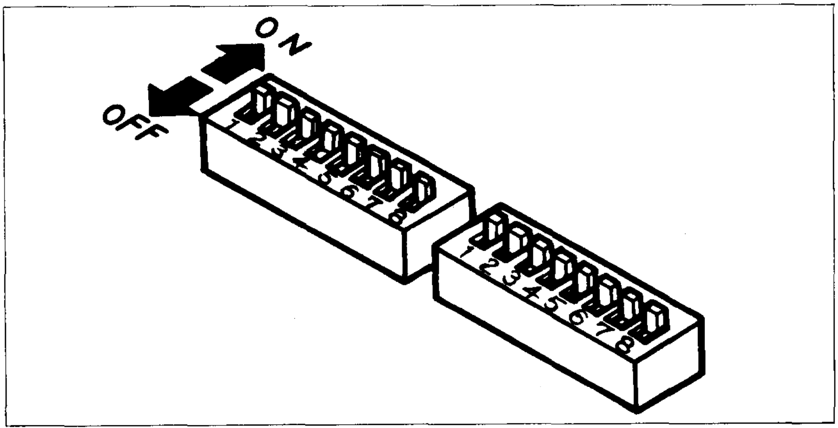


**Figure A-1.** A DIP switch is actually a series of several small switches.

All two DIP switches are readily accessible from the top. They are located in the compartment with the print head, and can be seen by opening the printer cover. To change the setting of a switch, use a ball-point pen or a similar object. The "on" position for a switch is towards the back of the printer; "off" is towards the front.

Never change the settings of any of the DIP switches when the power is on. Turn off both the printer and computer.

The individual switches on DIP switch 1 are called 1-1 through 1-8; those on switch 2 are 2-1 through 2-8.

Table A-1 summarizes the functions of DIP switches 1 and 2.

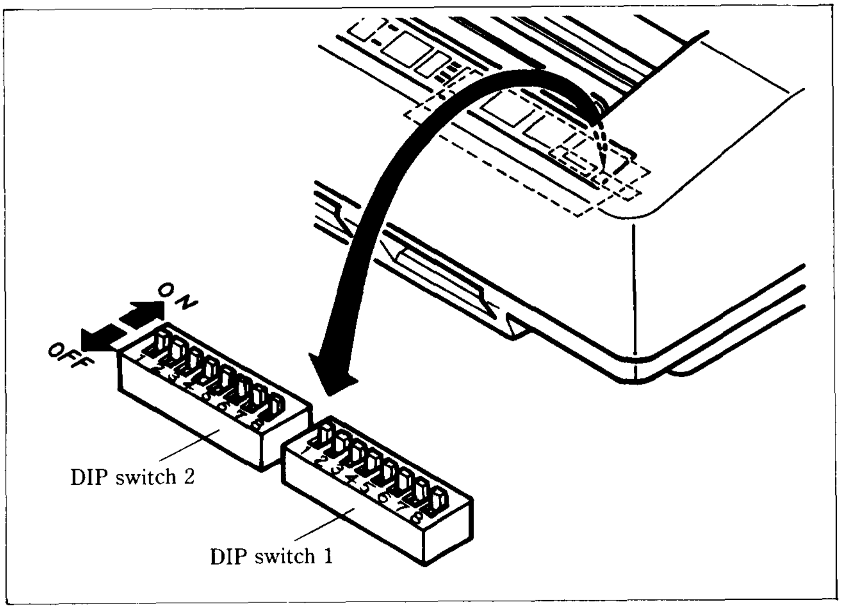


**Figure A-2** The DIP switches are located under the printer cover.

**Table A-1**
**DIP switch settings**

| Switch | ON | OFF |
|---|---|---|
| | **Switch 1** | |
| 1-1 | 10 CPI (pica pitch) | 17 CPI (condensed pitch) |
| 1-2 | Set SELECT IN signal to LOW | As is the SELECT IN signal |
| 1-3 | Select internal characters | Select optional characters |
| 1-4 | LQ character set | Draft character set |
| 1-5 | Character set #1 | Character set #2 |
| 1-6 | International character set selection — see Table A-2. | |
| 1-7 | | |
| 1-8 | | |
| | **Switch 2** | |
| 2-1 | No bottom margin | Set bottom margin to 1 inch |
| 2-2 | Set IBM mode | Set Standard mode |
| 2-3 | Ignore download characters | Enable download characters |
| 2-4 | Paper-out detected | Paper-out not detected |
| 2-5 | 8-bit interface | 7-bit interface |
| 2-6 | LF must be from host | Auto LF with CR |
| 2-7 | Print “normal zero” | Print “slash zero” |
| 2-8 | 1/6 inch line feed | 1/8 inch line feed |

## SWITCH FUNCTIONS

| Switch | Function |
|---|---|
| 1-1 | This switch selects the default character pitch. If this switch is on the default pitch is normal pica pitch (10 CPI). If this switch is off the default pitch is condensed pitch of pica (17 CPI). This switch is set on at the factory. |
| 1-2 | This switch controls the status of the SELECT IN signal of the parallel interface. If this switch is on, this signal is held to LOW. If this switch is off, the signal goes HIGH when the printer is not possible to get data. This switch is set on at the factory. |
| 1-3 | This switch selects the dafault character set. If this switch is on, the internal character set is selected as the default. If this switch is off the optional character set mounted on the Font 2 cartridge slot is selected (If the cartridge is not mounted, the internal character set is selected). This switch is set on at the factory. |
| 1-4 | This switch selects the default character style. If this switch is on the default character style is letter quality (LQ) characters. If this switch is off then the default character style is normal draft characters. This switch is set on at the factory. |
| 1-5 | This switch selects the default character set. If this switch is on the default character set is Character Set #1. If this switch is off the default character set is Character Set #2. If switch 2-2 is set off, this switch have no effect. This switch is set on at the factory. |
| 1-6~1-8 | These switches determine the default international character set, leaving the Japanese, Norwegian, and the second Danish, as shown in Table A-2. These switches are all set on at the factory. |

**Table A-2**
**International character sets**

| Switch | U.S.A | France | Germany | England | Denmark | Sweden | Italy | Spain |
|---|---|---|---|---|---|---|---|---|
| 1-6 | ON | OFF | ON | OFF | ON | OFF | ON | OFF |
| 1-7 | ON | ON | OFF | OFF | ON | ON | OFF | OFF |
| 1-8 | ON | ON | ON | ON | OFF | OFF | OFF | OFF |

2-1 This switch determines the default bottom margin. When this switch is on, the bottom margin is not set at the power-on. When this switch is off, the bottom margin is automatically set to 1 inch. This switch is set on at the factory.

2-2 This switch selects the active control codes. Turn this switch on to use the “IBM” compatible mode. Turn this switch off to use the “Standard” mode. This switch is set on at the factory.

2-3 This switch controls the RAM condition. When this switch is on, the download character definitions are ignored and the RAM is used as the print buffer. When this switch is off, the download character definitions are enable to use and the print buffer is set to one line buffer. This switch is set on at the factory.

2-4 This switch disables the paper-out detector. If the switch is on, the printer will signal the computer when it runs out of paper and will stop printing. If the switch is off, the printer will ignore the paper-out detector and will continue printing. This switch is set on at the factory.

2-5 This switch controls the eighth bit of the parallel interface. If this switch is on, the printer will read all eight bits on the parallel interface. If this switch is off, the printer will only read the first seven bits on the parallel interface and ignores the eighth bit. This switch is set on at the factory.

2-6 When this switch is on, the computer must send a line feed command every time the paper is to advance. When this switch is off, the printer will automatically advance the paper one line every time it receives a carriage return. (Most BASICs send a line feed with every carriage return, therefore, this switch should usually be on.) This switch is set on at the factory.

2-7 This switch selects the print style of “zeroes”. If this switch is on, the normal “zeroes” are printed. If this switch is off, the “slash zeroes” are printed. This switch is set on at the factory.

2-8 This switch sets the default line spacing. When this switch is on the default line spacing is set to 1/6 inch. This means that the printer will advance the paper 1/6 inch each time it receives a line feed. When this switch is off the default line spacing is 1/8 inch. This switch is set on at the factory.

# MEMO

# APPENDIX B

# ASCII CODES AND CONVERSION CHART

| Standard ASCII Codes | | | Control | Character set | |
|---|---|---|---|---|---|
| **Decimal** | **Hex.** | **Binary** | **Character** | **Set1** | **Set2** |
| 0 | 00 | 0000 0000 | Ctrl-@ | | |
| 1 | 01 | 0000 0001 | Ctrl-A | | |
| 2 | 02 | 0000 0010 | Ctrl-B | | |
| 3 | 03 | 0000 0011 | Ctrl-C | | ♥ ♥ |
| 4 | 04 | 0000 0100 | Ctrl-D | | ♦ ♦ |
| 5 | 05 | 0000 0101 | Ctrl-E | | ♣ ♣ |
| 6 | 06 | 0000 0110 | Ctrl-F | | ♠ ♠ |
| 7 | 07 | 0000 0111 | Ctrl-G | BEL | BEL |
| 8 | 08 | 0000 1000 | Ctrl-H | BS | BS |
| 9 | 09 | 0000 1001 | Ctrl-I | HT | HT |
| 10 | 0A | 0000 1010 | Ctrl-J | LF | LF |
| 11 | 0B | 0000 1011 | Ctrl-K | VT | VT |
| 12 | 0C | 0000 1100 | Ctrl-L | FF | FF |
| 13 | 0D | 0000 1101 | Ctrl-M | CR | CR |
| 14 | 0E | 0000 1110 | Ctrl-N | SO | SO |
| 15 | 0F | 0000 1111 | Ctrl-O | SI | SI |
| 16 | 10 | 0001 0000 | Ctrl-P | | |
| 17 | 11 | 0001 0001 | Ctrl-Q | DC1 | DC1 |
| 18 | 12 | 0001 0010 | Ctrl-R | DC2 | DC2 |
| 19 | 13 | 0001 0011 | Ctrl-S | DC3 | DC3 |
| 20 | 14 | 0001 0100 | Ctrl-T | DC4 | DC4 |
| 21 | 15 | 0001 0101 | Ctrl-U | | § § |
| 22 | 16 | 0001 0110 | Ctrl-V | | |
| 23 | 17 | 0001 0111 | Ctrl-W | | |
| 24 | 18 | 0001 1000 | Ctrl-X | CAN | CAN |
| 25 | 19 | 0001 1001 | Ctrl-Y | EM | EM |
| 26 | 1A | 0001 1010 | Ctrl-Z | | |
| 27 | 1B | 0001 1011 | | ESC | ESC |
| 28 | 1C | 0001 1100 | | | |
| 29 | 1D | 0001 1101 | | | |
| 30 | 1E | 0001 1110 | | | |
| 31 | 1F | 0001 1111 | | | |
| 32 | 20 | 0010 0000 | | Space | Space |

| | Standard ASCII Codes | | Character set | | | |
|---|---|---|---|---|---|---|
| Decimal | Hexadecimal | Binary | Set1 | | Set2 | |
| 33 | 21 | 0010 0001 | ! | ! | ! | ! |
| 34 | 22 | 0010 0010 | " | " | " | " |
| 35 | 23 | 0010 0011 | # | # | # | # |
| 36 | 24 | 0010 0100 | $ | $ | $ | $ |
| 37 | 25 | 0010 0101 | % | % | % | % |
| 38 | 26 | 0010 0110 | & | & | & | & |
| 39 | 27 | 0010 0111 | ' | ' | ' | ' |
| 40 | 28 | 0010 1000 | ( | ( | ( | ( |
| 41 | 29 | 0010 1001 | ) | ) | ) | ) |
| 42 | 2A | 0010 1010 | * | * | * | * |
| 43 | 2B | 0010 1011 | + | + | + | + |
| 44 | 2C | 0010 1100 | , | , | , | , |
| 45 | 2D | 0010 1101 | - | - | - | - |
| 46 | 2E | 0010 1110 | . | . | . | . |
| 47 | 2F | 0010 1111 | / | / | / | / |
| 48 | 30 | 0011 0000 | 0 | 0 | 0 | 0 |
| 49 | 31 | 0011 0001 | 1 | 1 | 1 | 1 |
| 50 | 32 | 0011 0010 | 2 | 2 | 2 | 2 |
| 51 | 33 | 0011 0011 | 3 | 3 | 3 | 3 |
| 52 | 34 | 0011 0100 | 4 | 4 | 4 | 4 |
| 53 | 35 | 0011 0101 | 5 | 5 | 5 | 5 |
| 54 | 36 | 0011 0110 | 6 | 6 | 6 | 6 |
| 55 | 37 | 0011 0111 | 7 | 7 | 7 | 7 |
| 56 | 38 | 0011 1000 | 8 | 8 | 8 | 8 |
| 57 | 39 | 0011 1001 | 9 | 9 | 9 | 9 |
| 58 | 3A | 0011 1010 | : | : | : | : |
| 59 | 3B | 0011 1011 | ; | ; | ; | ; |
| 60 | 3C | 0011 1100 | < | < | < | < |
| 61 | 3D | 0011 1101 | = | = | = | = |
| 62 | 3E | 0011 1110 | > | > | > | > |
| 63 | 3F | 0011 1111 | ? | ? | ? | ? |
| 64 | 40 | 0100 0000 | @ | @ | @ | @ |
| 65 | 41 | 0100 0001 | A | A | A | A |
| 66 | 42 | 0100 0010 | B | B | B | B |
| 67 | 43 | 0100 0011 | C | C | C | C |
| 68 | 44 | 0100 0100 | D | D | D | D |
| 69 | 45 | 0100 0101 | E | E | E | E |
| 70 | 46 | 0100 0110 | F | F | F | F |
| 71 | 47 | 0100 0111 | G | G | G | G |
| 72 | 48 | 0100 1000 | H | H | H | H |
| 73 | 49 | 0100 1001 | I | I | I | I |
| 74 | 4A | 0100 1010 | J | J | J | J |
| 75 | 4B | 0100 1011 | K | K | K | K |
| 76 | 4C | 0100 1100 | L | L | L | L |

| | Standard ASCII Codes | | Character set | | | |
|---|---|---|---|---|---|---|
| Decimal | Hexadecimal | Binary | Set1 | | Set2 | |
| 77 | 4D | 0100 1101 | M | M | M | M |
| 78 | 4E | 0100 1110 | N | N | N | N |
| 79 | 4F | 0100 1111 | O | O | O | O |
| 80 | 50 | 0101 0000 | P | P | P | P |
| 81 | 51 | 0101 0001 | Q | Q | Q | Q |
| 82 | 52 | 0101 0010 | R | R | R | R |
| 83 | 53 | 0101 0011 | S | S | S | S |
| 84 | 54 | 0101 0100 | T | T | T | T |
| 85 | 55 | 0101 0101 | U | U | U | U |
| 86 | 56 | 0101 0110 | V | V | V | V |
| 87 | 57 | 0101 0111 | W | W | W | W |
| 88 | 58 | 0101 1000 | X | X | X | X |
| 89 | 59 | 0101 1001 | Y | Y | Y | Y |
| 90 | 5A | 0101 1010 | Z | Z | Z | Z |
| 91 | 5B | 0101 1011 | [ | [ | [ | [ |
| 92 | 5C | 0101 1100 | \ | \ | \ | \ |
| 93 | 5D | 0101 1101 | ] | ] | ] | ] |
| 94 | 5E | 0101 1110 | ^ | ^ | ^ | ^ |
| 95 | 5F | 0101 1111 | _ | _ | _ | _ |
| 96 | 60 | 0110 0000 | ` | ` | ` | ` |
| 97 | 61 | 0110 0001 | a | a | a | a |
| 98 | 62 | 0110 0010 | b | b | b | b |
| 99 | 63 | 0110 0011 | c | c | c | c |
| 100 | 64 | 0110 0100 | d | d | d | d |
| 101 | 65 | 0110 0101 | e | e | e | e |
| 102 | 66 | 0110 0110 | f | f | f | f |
| 103 | 67 | 0110 0111 | g | g | g | g |
| 104 | 68 | 0110 1000 | h | h | h | h |
| 105 | 69 | 0110 1001 | i | i | i | i |
| 106 | 6A | 0110 1010 | j | j | j | j |
| 107 | 6B | 0110 1011 | k | k | k | k |
| 108 | 6C | 0110 1100 | l | l | l | l |
| 109 | 6D | 0110 1101 | m | m | m | m |
| 110 | 6E | 0110 1110 | n | n | n | n |
| 111 | 6F | 0110 1111 | o | o | o | o |
| 112 | 70 | 0111 0000 | p | p | p | p |
| 113 | 71 | 0111 0001 | q | q | q | q |
| 114 | 72 | 0111 0010 | r | r | r | r |
| 115 | 73 | 0111 0011 | s | s | s | s |
| 116 | 74 | 0111 0100 | t | t | t | t |
| 117 | 75 | 0111 0101 | u | u | u | u |
| 118 | 76 | 0111 0110 | v | v | v | v |
| 119 | 77 | 0111 0111 | w | w | w | w |
| 120 | 78 | 0111 1000 | x | x | x | x |

| | Standard ASCII Codes | | Character set | | | |
|---|---|---|---|---|---|---|
| Decimal | Hexadecimal | Binary | Set1 | | Set2 | |
| 121 | 79 | 0111 1001 | y | y | y | y |
| 122 | 7A | 0111 1010 | z | z | z | z |
| 123 | 7B | 0111 1011 | { | { | { | { |
| 124 | 7C | 0111 1100 | ¦ | ¦ | ¦ | ¦ |
| 125 | 7D | 0111 1101 | } | } | } | } |
| 126 | 7E | 0111 1110 | ~ | ~ | ~ | ~ |
| 127 | 7F | 0111 1111 | DEL | | DEL | |
| 128 | 80 | 1000 0000 | | | Ç | Ç |
| 129 | 81 | 1000 0001 | | | ü | ü |
| 130 | 82 | 1000 0010 | | | é | é |
| 131 | 83 | 1000 0011 | | | â | â |
| 132 | 84 | 1000 0100 | | | ä | ä |
| 133 | 85 | 1000 0101 | | | à | à |
| 134 | 86 | 1000 0110 | | | å | å |
| 135 | 87 | 1000 0111 | BEL | | ç | ç |
| 136 | 88 | 1000 1000 | BS | | ê | ê |
| 137 | 89 | 1000 1001 | HT | | ë | ë |
| 138 | 8A | 1000 1010 | LF | | è | è |
| 139 | 8B | 1000 1011 | VT | | ï | ï |
| 140 | 8C | 1000 1100 | FF | | î | î |
| 141 | 8D | 1000 1101 | CR | | ì | ì |
| 142 | 8E | 1000 1110 | SO | | Ä | Ä |
| 143 | 8F | 1000 1111 | SI | | Å | Å |
| 144 | 90 | 1001 0000 | | | É | É |
| 145 | 91 | 1001 0001 | DC1 | | æ | æ |
| 146 | 92 | 1001 0010 | DC2 | | Æ | Æ |
| 147 | 93 | 1001 0011 | DC3 | | ô | ô |
| 148 | 94 | 1001 0100 | DC4 | | ö | ö |
| 149 | 95 | 1001 0101 | | | ò | ò |
| 150 | 96 | 1001 0110 | | | û | û |
| 151 | 97 | 1001 0111 | | | ù | ù |
| 152 | 98 | 1001 1000 | CAN | | ÿ | ÿ |
| 153 | 99 | 1001 1001 | EM | | Ö | Ö |
| 154 | 9A | 1001 1010 | | | Ü | Ü |
| 155 | 9B | 1001 1011 | ESC | | ¢ | ¢ |
| 156 | 9C | 1001 1100 | | | £ | £ |
| 157 | 9D | 1001 1101 | | | ¥ | ¥ |
| 158 | 9E | 1001 1110 | | | ₧ | ₧ |
| 159 | 9F | 1001 1111 | | | ƒ | ƒ |
| 160 | A0 | 1010 0000 | á | á | á | á |
| 161 | A1 | 1010 0001 | í | í | í | í |
| 162 | A2 | 1010 0010 | ó | ó | ó | ó |
| 163 | A3 | 1010 0011 | ú | ú | ú | ú |
| 164 | A4 | 1010 0100 | ñ | ñ | ñ | ñ |

| | Standard ASCII Codes | | Character set | | | |
|---|---|---|---|---|---|---|
| Decimal | Hexadecimal | Binary | Set1 | | Set2 | |
| 165 | A5 | 1010 0101 | Ñ | Ñ | Ñ | Ñ |
| 166 | A6 | 1010 0110 | ª | ª | ª | ª |
| 167 | A7 | 1010 0111 | º | º | º | º |
| 168 | A8 | 1010 1000 | ¿ | ¿ | ¿ | ¿ |
| 169 | A9 | 1010 1001 | ⌐ | ⌐ | ⌐ | ⌐ |
| 170 | AA | 1010 1010 | ¬ | ¬ | ¬ | ¬ |
| 171 | AB | 1010 1011 | ½ | ½ | ½ | ½ |
| 172 | AC | 1010 1100 | ¼ | ¼ | ¼ | ¼ |
| 173 | AD | 1010 1101 | ¡ | ¡ | ¡ | ¡ |
| 174 | AE | 1010 1110 | « | « | « | « |
| 175 | AF | 1010 1111 | » | » | » | » |
| 176 | B0 | 1011 0000 | ░ | ░ | ░ | ░ |
| 177 | B1 | 1011 0001 | ▒ | ▒ | ▒ | ▒ |
| 178 | B2 | 1011 0010 | ▓ | ▓ | ▓ | ▓ |
| 179 | B3 | 1011 0011 | │ | │ | │ | │ |
| 180 | B4 | 1011 0100 | ┤ | ┤ | ┤ | ┤ |
| 181 | B5 | 1011 0101 | ╡ | ╡ | ╡ | ╡ |
| 182 | B6 | 1011 0110 | ╢ | ╢ | ╢ | ╢ |
| 183 | B7 | 1011 0111 | ╖ | ╖ | ╖ | ╖ |
| 184 | B8 | 1011 1000 | ╕ | ╕ | ╕ | ╕ |
| 185 | B9 | 1011 1001 | ╣ | ╣ | ╣ | ╣ |
| 186 | BA | 1011 1010 | ║ | ║ | ║ | ║ |
| 187 | BB | 1011 1011 | ╗ | ╗ | ╗ | ╗ |
| 188 | BC | 1011 1100 | ╝ | ╝ | ╝ | ╝ |
| 189 | BD | 1011 1101 | ╜ | ╜ | ╜ | ╜ |
| 190 | BE | 1011 1110 | ╛ | ╛ | ╛ | ╛ |
| 191 | BF | 1011 1111 | ┐ | ┐ | ┐ | ┐ |
| 192 | C0 | 1100 0000 | └ | └ | └ | └ |
| 193 | C1 | 1100 0001 | ┴ | ┴ | ┴ | ┴ |
| 194 | C2 | 1100 0010 | ┬ | ┬ | ┬ | ┬ |
| 195 | C3 | 1100 0011 | ├ | ├ | ├ | ├ |
| 196 | C4 | 1100 0100 | ─ | ─ | ─ | ─ |
| 197 | C5 | 1100 0101 | ┼ | ┼ | ┼ | ┼ |
| 198 | C6 | 1100 0110 | ╞ | ╞ | ╞ | ╞ |
| 199 | C7 | 1100 0111 | ╟ | ╟ | ╟ | ╟ |
| 200 | C8 | 1100 1000 | ╚ | ╚ | ╚ | ╚ |
| 201 | C9 | 1100 1001 | ╔ | ╔ | ╔ | ╔ |
| 202 | CA | 1100 1010 | ╩ | ╩ | ╩ | ╩ |

| | Standard ASCII Codes | | Character set | | | |
|---|---|---|---|---|---|---|
| Decimal | Hexadecimal | Binary | Set1 | | Set2 | |
| 203 | C B | 1100 1011 | ╦ | ╦ | ╦ | ╦ |
| 204 | C C | 1100 1100 | ╠ | ╠ | ╠ | ╠ |
| 205 | C D | 1100 1101 | ═ | ═ | ═ | ═ |
| 206 | C E | 1100 1110 | ╬ | ╬ | ╬ | ╬ |
| 207 | C F | 1100 1111 | ╧ | ╧ | ╧ | ╧ |
| 208 | D0 | 1101 0000 | ╨ | ╨ | ╨ | ╨ |
| 209 | D1 | 1101 0001 | ╤ | ╤ | ╤ | ╤ |
| 210 | D2 | 1101 0010 | ╥ | ╥ | ╥ | ╥ |
| 211 | D3 | 1101 0011 | ╙ | ╙ | ╙ | ╙ |
| 212 | D4 | 1101 0100 | ╘ | ╘ | ╘ | ╘ |
| 213 | D5 | 1101 0101 | ╒ | ╒ | ╒ | ╒ |
| 214 | D6 | 1101 0110 | ╓ | ╓ | ╓ | ╓ |
| 215 | D7 | 1101 0111 | ╫ | ╫ | ╫ | ╫ |
| 216 | D8 | 1101 1000 | ╪ | ╪ | ╪ | ╪ |
| 217 | D9 | 1101 1001 | ┘ | ┘ | ┘ | ┘ |
| 218 | D A | 1101 1010 | ┌ | ┌ | ┌ | ┌ |
| 219 | D B | 1101 1011 | █ | █ | █ | █ |
| 220 | D C | 1101 1100 | ▄ | ▄ | ▄ | ▄ |
| 221 | D D | 1101 1101 | ▌ | ▌ | ▌ | ▌ |
| 222 | D E | 1101 1110 | ▐ | ▐ | ▐ | ▐ |
| 223 | D F | 1101 1111 | ▀ | ▀ | ▀ | ▀ |
| 224 | E0 | 1110 0000 | α | α | α | α |
| 225 | E1 | 1110 0001 | β | β | β | β |
| 226 | E2 | 1110 0010 | Γ | Γ | Γ | Γ |
| 227 | E3 | 1110 0011 | π | π | π | π |
| 228 | E4 | 1110 0100 | Σ | Σ | Σ | Σ |
| 229 | E5 | 1110 0101 | σ | σ | σ | σ |
| 230 | E6 | 1110 0110 | μ | μ | μ | μ |
| 231 | E7 | 1110 0111 | τ | τ | τ | τ |
| 232 | E8 | 1110 1000 | Φ | Φ | Φ | Φ |
| 233 | E9 | 1110 1001 | Θ | Θ | Θ | Θ |
| 234 | E A | 1110 1010 | Ω | Ω | Ω | Ω |
| 235 | E B | 1110 1011 | δ | δ | δ | δ |
| 236 | E C | 1110 1100 | ∞ | ∞ | ∞ | ∞ |
| 237 | E D | 1110 1101 | φ | φ | φ | φ |
| 238 | E E | 1110 1110 | ∈ | ∈ | ∈ | ∈ |
| 239 | E F | 1110 1111 | ∩ | ∩ | ∩ | ∩ |
| 240 | F0 | 1111 0000 | ≡ | ≡ | ≡ | ≡ |
| 241 | F1 | 1111 0001 | ± | ± | ± | ± |

| | Standard ASCII Codes | | Character set | | | |
|---|---|---|---|---|---|---|
| Decimal | Hexadecimal | Binary | Set1 | | Set2 | |
| 242 | F 2 | 1111 0010 | ≥ | ≥ | ≥ | ≥ |
| 243 | F 3 | 1111 0011 | ≤ | ≤ | ≤ | ≤ |
| 244 | F 4 | 1111 0100 | ⌠ | ⌠ | ⌠ | ⌠ |
| 245 | F 5 | 1111 0101 | ⌡ | ⌡ | ⌡ | ⌡ |
| 246 | F 6 | 1111 0110 | ÷ | ÷ | ÷ | ÷ |
| 247 | F 7 | 1111 0111 | ≈ | ≈ | ≈ | ≈ |
| 248 | F 8 | 1111 1000 | ° | ° | ° | ° |
| 249 | F 9 | 1111 1001 | ∙ | ∙ | ∙ | ∙ |
| 250 | F A | 1111 1010 | - | - | - | - |
| 251 | F B | 1111 1011 | √ | √ | √ | √ |
| 252 | F C | 1111 1100 | ⁿ | ⁿ | ⁿ | ⁿ |
| 253 | F D | 1111 1101 | ² | ² | ² | ² |
| 254 | F E | 1111 1110 | ■ | ■ | ■ | ■ |
| 255 | F F | 1111 1111 | | | | |

# MEMO

# APPENDIX C

# FUNCTION CODES

The purpose of this Appendix is to provide a quick reference for the various functions available on this printer. Codes are described in the following format.

| | |
|---|---|
| PURPOSE | **Tells what the function code does.** |
| CODE | Control code mnemonic |
| (decimal ASCII) | ASCII decimal equivalent |
| (hex ASCII) | Hexadecimal equivalent |
| REMARKS | Briefly describes how the command is used. |
| SEE | Tells where details of the command may be found. |

Several commands require you to specify a value or values. In these cases, we have used an "*n*" or "*m*" to indicate a variable. You should insert the ASCII code for the proper value here.

## COMMANDS TO CONTROL PRINT STYLE

These commands are used to control the font style, the print pitch, and special effects.

■ Font style controls

PURPOSE **Selects the LQ characters.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | “x” | 1 |
| (decimal ASCII) | 27 | 120 | 1 |
| (hex ASCII) | 1B | 78 | 01 |

REMARKS This command causes the printer to print letter quality (LQ) characters until LQ mode is cancelled. LQ mode cannot be used with any other special printing functions except underlining, overlining, expanded printing, and big character printing. You can select the LQ characters at the power-on default by turning DIP switch 1-4 on. This command is ignored when the “Panel” mode is selected at the power-on.

***Note:*** The character “1” (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.

SEE Chapter 5

| | | | |
|---|---|---|---|
| PURPOSE | **Cancels the LQ characters.** | | |
| CODE | ⟨ESC⟩ | "x" | 0 |
| (decimal ASCII) | 27 | 120 | 0 |
| (hex ASCII) | 1B | 78 | 00 |
| REMARKS | This command cancels the LQ printing and returns the printer to the draft mode. You can select the draft characters as the power-on default by turning DIP switch 1-4 off. This command is ignored when the "Panel" mode is selected at the power-on.<br>***Note:*** The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0. | | |
| SEE | Chapter 5 | | |

| | | |
|---|---|---|
| PURPOSE | **Selects character set #1.** | |
| CODE | ⟨ESC⟩ | "7" |
| (decimal ASCII) | 27 | 55 |
| (hex ASCII) | 1B | 37 |
| REMARKS | This command causes the printer to cancel character set #2 and selects instead character set #1 when the DIP switch 2-2 is set on. You can select character set #1 as the power-on default by turning DIP switch 1-5 on while the DIP switch 2-2 is set on. | |
| SEE | Chapter 7 | |

| | | |
|---|---|---|
| PURPOSE | **Selects character set #2.** | |
| CODE | ⟨ESC⟩ | "6" |
| (decimal ASCII) | 27 | 54 |
| (hex ASCII) | 1B | 36 |
| REMARKS | This command selects character set #2 when the DIP switch 2-2 is set on. You can select character set #2 as the power-on default by turning DIP switch 1-5 off while the DIP switch 2-2 is set on. | |
| SEE | Chapter 7 | |

PURPOSE **Selects an international character set.**

| | | | |
|---|---|---|---|
| CODE | 〈ESC〉 | "R" | *n* |
| (decimal ASCII) | 27 | 82 | *n* |
| (hex ASCII) | 1B | 52 | *n* |

REMARKS This command selects the international character set according to the value of *n* as shown in the table below:

| *n* | Character set | *n* | Character set |
|---|---|---|---|
| 0 | U.S.A | 6 | Italy |
| 1 | France | 7 | Spain |
| 2 | Germany | 8 | Japan |
| 3 | England | 9 | Norway |
| 4 | Denmark Ⅰ | 10 | Denmark Ⅱ |
| 5 | Sweden | | |

You can select a particular international character set, except Japan, Norway, and Denmark type Ⅱ, as a power-on default by adjusting the settings of DIP switches 1-6, 1-7, and 1-8.

SEE Chapter 7

PURPOSE **Selects a character set.**

| | | | |
|---|---|---|---|
| CODE | 〈ESC〉 | "k" | *n* |
| (decimal ASCII) | 27 | 107 | *n* |
| (hex ASCII) | 1B | 6B | *n* |

REMARKS This command selects one of the character sets mounted on the printer depending the value of *n*. When the value of *n* is 0 then the character set is selected the internal character set. When *n* is 1 it is selected the character set mounted on the Font 1 slot. When *n* is 2 it is selected the character set mounted on the Font 2 slot. This command is ignored when the "Panel" mode is selected at the power-on.

SEE Chapter 5

| | | |
|---|---|---|
| PURPOSE | **Selects optional character set.** | |
| CODE | 〈ESC〉 | "4" |
| (decimal ASCII) | 27 | 52 |
| (hex ASCII) | 1B | 34 |
| REMARKS | This command selects the character set mounted on the Font 1 slot of the printer. If the font cartridge is not mounted or the "Panel" mode is selected at the power-on, this command is ignored. | |
| SEE | Chapter 5 | |

| | | |
|---|---|---|
| PURPOSE | **Cançels optional character set.** | |
| CODE | 〈ESC〉 | "5" |
| (decimal ASCII) | 27 | 53 |
| (hex ASCII) | 1B | 35 |
| REMARKS | This command cancels the optional character set with 〈ESC〉"4", and returns to the previous character set. This command is ignored when the "Panel" mode is selected at the power-on. | |
| SEE | Chapter 5 | |

■ Font pitch controls

| | | |
|---|---|---|
| PURPOSE | **Sets the print pitch to pica.** | |
| CODE | 〈ESC〉 | "P" |
| (decimal ASCII) | 27 | 80 |
| (hex ASCII) | 1B | 50 |
| REMARKS | This command causes printing to be done in pica pitch with 136 characters per line. You can select the pica pitch as the power-on default by turning DIP switch 1-1 on. This command is ignored when the "Panel" mode is selected at the power-on. | |
| SEE | Chapter 5 | |

| | | |
|---|---|---|
| PURPOSE | **Sets the print pitch to elite.** | |
| CODE | ⟨ESC⟩ | "M" |
| (decimal ASCII) | 27 | 77 |
| (hex ASCII) | 1B | 4D |
| REMARKS | This command causes printing to be done in elite pitch with 163 characters per line. This command is ignored when the "Panel" mode is selected at the power-on. | |
| SEE | Chapter 5 | |

| | | |
|---|---|---|
| PURPOSE | **Sets the printer to condensed print.** | |
| CODE | ⟨SI⟩ | |
| (decimal ASCII) | 15 | |
| (hex ASCII) | 0F | |
| REMARKS | This command causes printing to be done in condensed pitch with 222 characters per line for pica condensed, and 234 characters per line for elite condensed (LQ characters are not printed in condensed pitch). You can select the pica condensed pitch with the control panel, but you cannot select the elite condensed pitch manually. This command is ignored when the "Panel" mode is selected at the power-on. | |
| SEE | Chapter 5 | |

| | | |
|---|---|---|
| PURPOSE | **Sets the printer to condensed print.** | |
| CODE | ⟨ESC⟩ | ⟨SI⟩ |
| (decimal ASCII) | 27 | 15 |
| (hex ASCII) | 1B | 0F |
| REMARKS | Same as ⟨SI⟩, above. | |
| SEE | Chapter 5 | |

PURPOSE **Cancels the condensed print.**

| | |
|---|---|
| CODE | 〈DC2〉 |
| (decimal ASCII) | 18 |
| (hex ASCII) | 12 |

REMARKS This command cancels the condensed printing and returns the printer to the normal print pitch. This command is ignored when the "Panel" mode is selected at the power-on.

SEE Chapter 5

PURPOSE **Sets the printer to expanded print.**

| | | | |
|---|---|---|---|
| CODE | 〈ESC〉 | "W" | 1 |
| (decimal ASCII) | 27 | 87 | 1 |
| (hex ASCII) | 1B | 57 | 01 |

REMARKS This command causes characters to be printed twice as wide as normally (half the current pitch) until expanded printing is cancelled.
***Note:*** The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.

SEE Chapter 5

PURPOSE **Cancels the expanded print.**

| | | | |
|---|---|---|---|
| CODE | 〈ESC〉 | "W" | 0 |
| (decimal ASCII) | 27 | 87 | 0 |
| (hex ASCII) | 1B | 57 | 00 |

REMARKS This command resets the character pitch to what it was before expanded printing was set.
***Note:*** The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.

SEE Chapter 5

| | |
|---|---|
| PURPOSE | **Sets the printer to expanded print for the remainder of the current line.** |
| CODE | 〈SO〉 |
| (decimal ASCII) | 14 |
| (hex ASCII) | 0E |
| REMARKS | This command causes characters to be printed twice as wide as normally until a carriage return is sent. It also cancelled with 〈DC4〉. |
| SEE | Chapter 5 |

| | | |
|---|---|---|
| PURPOSE | **Sets the printer to expanded print for the remainder of the current line.** | |
| CODE | 〈ESC〉 | 〈SO〉 |
| (decimal ASCII) | 27 | 14 |
| (hex ASCII) | 1B | 0E |
| REMARKS | Same as 〈SO〉, above. | |
| SEE | Chapter 5 | |

| | |
|---|---|
| PURPOSE | **Cancels one line expanded print.** |
| CODE | 〈DC4〉 |
| (decimal ASCII) | 20 |
| (hex ASCII) | 14 |
| REMARKS | This command cancels one line expanded print set with 〈SO〉 or 〈ESC〉 〈SO〉. |
| SEE | Chapter 5 |

PURPOSE **Sets the printer to proportional print.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | “p” | 1 |
| (decimal ASCII) | 27 | 112 | 1 |
| (hex ASCII) | 1B | 70 | 01 |

REMARKS This command causes characters to be printed with proportional spacing until proportional printing is cancelled. This command is ignored when the “Panel” mode is selected at the power-on.
***Note:*** The character “1” (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.

SEE Chapter 5

PURPOSE **Cancels proportional print.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | “p” | 0 |
| (decimal ASCII) | 27 | 112 | 0 |
| (hex ASCII) | 1B | 70 | 00 |

REMARKS This command cancels the proportional printing and returns to the “fixed pitch” printing. This command is ignored when the “Panel” mode is selected at the power-on.
***Note:*** The character “0” (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.

SEE Chapter 5

## ■ Special print modes

| | | | |
|---|---|---|---|
| PURPOSE | **Sets the master print mode.** | | |
| CODE | ⟨ESC⟩ | "!" | *n* |
| (decimal ASCII) | 27 | 33 | *n* |
| (hex ASCII) | 1B | 21 | *n* |
| REMARKS | This is a powerful command that allows the user to set several printing characteristics at one time: print pitch, condensed print, expanded print, boldface, underlining, and any combination of these as determined by *n*, a number from 0 to 255. (See Table 5-10 for details.) | | |
| SEE | Chapter 5 | | |

| | | |
|---|---|---|
| PURPOSE | **Selects boldface printing.** | |
| CODE | ⟨ESC⟩ | "E" |
| (decimal ASCII) | 27 | 69 |
| (hex ASCII) | 1B | 45 |
| REMARKS | This command causes characters to be printed in boldface until boldface is cancelled. You cannot be used with superscripts, or subscripts in boldface printing. | |
| SEE | Chapter 5 | |

| | | |
|---|---|---|
| PURPOSE | **Selects boldface printing.** | |
| CODE | ⟨ESC⟩ | "G" |
| (decimal ASCII) | 27 | 71 |
| (hex ASCII) | 1B | 47 |
| REMARKS | Same as ⟨ESC⟩ "E", above. | |
| SEE | Chapter 5 | |

| | | | |
|---|---|---|---|
| PURPOSE | **Cancels boldface printing.** | | |
| CODE | ⟨ESC⟩ | "F" | |
| (decimal ASCII) | 27 | 70 | |
| (hex ASCII) | 1B | 46 | |
| REMARKS | This command cancels boldface printing and returns the printer to normal printing. | | |
| SEE | Chapter 5 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Cancels boldface printing.** | | |
| CODE | ⟨ESC⟩ | "H" | |
| (decimal ASCII) | 27 | 72 | |
| (hex ASCII) | 1B | 48 | |
| REMARKS | Same as ⟨ESC⟩ "F", above. | | |
| SEE | Chapter 5 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Selects underlining.** | | |
| CODE | ⟨ESC⟩ | "–" | 1 |
| (decimal ASCII) | 27 | 45 | 1 |
| (hex ASCII) | 1B | 2D | 01 |
| REMARKS | This command underlines the following characters until underlining is cancelled. ***Note:*** The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1. | | |
| SEE | Chapter 5 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Cancels underlining.** | | |
| CODE | ⟨ESC⟩ | "–" | 0 |
| (decimal ASCII) | 27 | 45 | 0 |
| (hex ASCII) | 1B | 2D | 00 |
| REMARKS | This command stops underlining. ***Note:*** The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0. | | |
| SEE | Chapter 5 | | |

PURPOSE **Selects overlining.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | “_” | 1 |
| (decimal ASCII) | 27 | 95 | 1 |
| (hex ASCII) | 1B | 5F | 01 |

REMARKS This command overlines the following characters until overlining is cancelled. ***Note:*** The character “1” (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.

SEE Chapter 5

PURPOSE **Cancels overlining.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | “_” | 0 |
| (decimal ASCII) | 27 | 95 | 0 |
| (hex ASCII) | 1B | 5F | 00 |

REMARKS This command stops overlining. ***Note:*** The character “0” (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.

SEE Chapter 5

PURPOSE **Selects superscripts.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | “S” | 0 |
| (decimal ASCII) | 27 | 83 | 0 |
| (hex ASCII) | 1B | 53 | 00 |

REMARKS This command raises the following characters and prints them as superscripts until superscripting is cancelled. Superscripts are not printed as elite pitch or condensed pitch. They are always printed with pica or expanded pica pitch. ***Note:*** The character “0” (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.

SEE Chapter 5

| | | | |
|---|---|---|---|
| PURPOSE | **Selects subscripts.** | | |
| CODE | ⟨ESC⟩ | "S" | 1 |
| (decimal ASCII) | 27 | 83 | 1 |
| (hex ASCII) | 1B | 53 | 01 |
| REMARKS | This command lowers the following characters and prints as subscripts until subscripting is cancelled. All conditions described for superscripts also apply to subscripts.<br>***Note:*** The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1. | | |
| SEE | Chapter 5 | | |

| | | |
|---|---|---|
| PURPOSE | **Cancels a superscript or subscript.** | |
| CODE | ⟨ESC⟩ | "T" |
| (decimal ASCII) | 27 | 84 |
| (hex ASCII) | 1B | 54 |
| REMARKS | This command stops printing of superscripts or subscripts and returns to the normal printing previously set. | |
| SEE | Chapter 5 | |

## CONTROLLING THE VERTICAL PRINT POSITION

These commands are used to move the paper relative to the print head. By moving the paper up or down, the print head, in effect, moves the opposite direction (down or up) on the page.

■ Line feed controls

| | | |
|---|---|---|
| PURPOSE | **Advances the paper one line (line feed).** | |
| CODE | ⟨LF⟩ | |
| (decimal ASCII) | 10 | |
| (hex ASCII) | 0A | |
| REMARKS | The actual distance advanced by the line feed is set either through DIP switch 2-8 or through various codes which can be sent (see below). When the DIP switch 2-6 is off, a line feed is automatically generated whenever the printer receives a carriage return. | |
| SEE | Chapter 6 | |

| | | |
|---|---|---|
| PURPOSE | **Sets line spacing to 1/8 inch.** | |
| CODE | ⟨ESC⟩ | "0" |
| (decimal ASCII) | 27 | 48 |
| (hex ASCII) | 1B | 30 |
| REMARKS | This command sets the actual distance the paper advances during all subsequent line feeds to 1/8 inch. You can select 1/8 inch line spacing as the power-on default by turning DIP switch 2-8 off. | |
| SEE | Chapter 6 | |

PURPOSE **Sets line spacing to 7/60 inch.**

| | | |
|---|---|---|
| CODE | ⟨ESC⟩ | "1" |
| (decimal ASCII) | 27 | 49 |
| (hex ASCII) | 1B | 31 |

REMARKS This command sets the actual distance the paper advances during all subsequent line feeds to 7/60 inch.

SEE Chapter 6

PURPOSE **Sets line spacing to *n*/180 inch.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | "3" | *n* |
| (decimal ASCII) | 27 | 51 | *n* |
| (hex ASCII) | 1B | 33 | *n* |

REMARKS This command sets the actual distance the paper advances during all subsequent line feeds to *n*/180 inch. The value of *n* must be between 1 and 255.

SEE Chapter 6

PURPOSE **Sets or defines line spacing to *n*/60 inch.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | "A" | *n* |
| (decimal ASCII) | 27 | 65 | *n* |
| (hex ASCII) | 1B | 41 | *n* |

REMARKS This command works in two different functions depending on the setting of DIP switch 2-2. When the DIP switch 2-2 is set on, this command defines the actual distance the paper advances during all subsequent line feeds to *n*/60 inch. This command must be used in conjunction with ⟨ESC⟩ "2" which activates the ⟨ESC⟩ "A" definition.

When the DIP switch 2-2 is set off, this command sets the actual distance the paper advances during all subsequent line feeds to *n*/60 inch immediately. The value of *n* must be between 1 and 255.

SEE Chapter 6

| | | | |
|---|---|---|---|
| PURPOSE | **Sets line spacing to 1/6 inch, or Uses ⟨ESC⟩"A" definition.** | | |
| CODE | ⟨ESC⟩ | "2" | |
| (decimal ASCII) | 27 | 50 | |
| (hex ASCII) | 1B | 32 | |
| REMARKS | This command works in two different functions depending on the setting of DIP switch 2-2. When the DIP switch 2-2 is set on, this command activates the line spacing defined in the ⟨ESC⟩ "A" command. If the ⟨ESC⟩ "A" command has not been defined, the line spacing is changed to 1/6 inch. When the DIP switch 2-2 is set off, this command sets the actual distance the paper advances during all subsequent line feeds to 1/6 inch. You can select the 1/6 inch line spacing as the power-on default by turning DIP switch 2-8 off. | | |
| SEE | Chapter 6 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Sends a one-time paper feed of *n*/180 inch.** | | |
| CODE | ⟨ESC⟩ | "J" | *n* |
| (decimal ASCII) | 27 | 74 | *n* |
| (hex ASCII) | 1B | 4A | *n* |
| REMARKS | This command causes the printer to advance the paper *n*/180 inch. It does not change the current value of line spacing and it does not cause a carriage return. The value of *n* must be between 1 and 255. | | |
| SEE | Chapter 6 | | |

PURPOSE **Sends a one-time reverse feed of *n*/180 inch.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | "j" | *n* |
| (decimal ASCII) | 27 | 106 | *n* |
| (hex ASCII) | 1B | 6A | *n* |

REMARKS This command causes the printer to reverse the paper *n*/180 inch. It does not change the current value of line spacing and it does not cause a carriage return. The value of *n* must be between 1 and 255.

SEE Chapter 6

PURPOSE **Sets print position to *n* lines.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "f" | 1 | *n* |
| (decimal ASCII) | 27 | 102 | 1 | *n* |
| (hex ASCII) | 1B | 66 | 01 | *n* |

REMARKS This command sets the next print position to the *n*th line from the top of the current page.

***Note:*** The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.

SEE Chapter 6

■ Form feed and related commands

PURPOSE **Advances the paper to the top of the next page (form feed).**

| | |
|---|---|
| CODE | 〈FF〉 |
| (decimal ASCII) | 12 |
| (hex ASCII) | 0C |

REMARKS The actual length of a page ejected by a form feed is set either by setting of the control panel key or through various codes which can be sent (see below). This command works as the ejecting paper command when the optional automatic sheet feeder is mounted on the printer.

SEE Chapter 6

PURPOSE **Sets page length to *n* inches.**

| | | | | |
|---|---|---|---|---|
| CODE | 〈ESC〉 | “C” | 0 | *n* |
| (decimal ASCII) | 27 | 67 | 0 | *n* |
| (hex ASCII) | 1B | 43 | 00 | *n* |

REMARKS This command sets the length of all subsequent pages to *n* inches. The value of *n* must be between 1 and 22. You can select a power-on default form length by setting the form length dial on the control panel. This command is ignored when the optional automatic sheet feeder is mounted on the printer.

SEE Chapter 6

| | | | |
|---|---|---|---|
| PURPOSE | **Sets page length to *n* lines.** | | |
| CODE | 〈ESC〉 | “C” | *n* |
| (decimal ASCII) | 27 | 67 | *n* |
| (hex ASCII) | 1B | 43 | *n* |
| REMARKS | This command sets the length of all subsequent pages to *n* lines. The value of *n* must be between 1 and 255. This command is ignored when the optional automatic sheet feeder is mounted on the printer. | | |
| SEE | Chapter 6 | | |

■ Bottom margin and vertical tabs

| | | | |
|---|---|---|---|
| PURPOSE | **Sets the bottom margin.** | | |
| CODE | 〈ESC〉 | “N” | *n* |
| (decimal ASCII) | 27 | 78 | *n* |
| (hex ASCII) | 1B | 4E | *n* |
| REMARKS | This command sets the bottom margin to *n* lines. The printer will generate a form feed whenever there are *n* lines left on the page. This command is ignored when the optional automatic sheet feeder is mounted on the printer. The value of *n* must be between 1 and 127. | | |
| SEE | Chapter 6 | | |

| | | |
|---|---|---|
| PURPOSE | **Cancels the bottom margin.** | |
| CODE | 〈ESC〉 | “O” |
| (decimal ASCII) | 27 | 79 |
| (hex ASCII) | 1B | 4F |
| REMARKS | This command cancels the bottom margin set by 〈ESC〉 “N” command. | |
| SEE | Chapter 6 | |

PURPOSE **Advances paper to the next vertical tab position.**

| | |
|---|---|
| CODE | ⟨VT⟩ |
| (decimal ASCII) | 11 |
| (hex ASCII) | 0B |

REMARKS This command causes the paper to be advanced to the next vertical tab position, or the top of the next page, whichever it finds first. If the vertical tab positions are not set, this command works as a line feed command.

SEE Chapter 6

PURPOSE **Sets vertical tab positions.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "B" | *n1 n2 n3* ... | 0 |
| (decimal ASCII) | 27 | 66 | *n1 n2 n3* ... | 0 |
| (hex ASCII) | 1B | 42 | *n1 n2 n3* ... | 00 |

REMARKS This command cancels all current vertical tab positions and sets those defined at lines *n1, n2, n3,* etc. The maximum number of vertical tab positions allowed is 16. The ASCII 0 character is used as a command terminator. Each vertical tab position must be specified in ascending order.

SEE Chapter 6

PURPOSE **Selects vertical channel.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | "/" | *n0* |
| (decimal ASCII) | 27 | 47 | *n0* |
| (hex ASCII) | 1B | 2F | *n0* |

REMARKS This command selects one of the multiple vertical channels determined by the value of *n0*. The value of *n0* must be between 0 and 7.

SEE Chapter 6

PURPOSE

**Sets vertical tab positions in a channel.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "b" | *n0 n1 n2 n3* ... | 0 |
| (decimal ASCII) | 27 | 98 | *n0 n1 n2 n3* ... | 0 |
| (hex ASCII) | 1B | 62 | *n0 n1 n2 n3* ... | 00 |

REMARKS

This command cancels all current vertical tab positions in channel *n0* and sets those defined at lines *n1, n2, n3,* etc. The maximum number of vertical tab positions for each channel allowed is 16. The ASCII 0 character is used as a command terminator. Each vertical tab position must be specified in ascending order. The vertical channel, *n0*, must be between 0 and 7.

SEE

Chapter 6

PURPOSE

**Sets vertical tab positions every *n* lines.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "e" | 1 | *n* |
| (decimal ASCII) | 27 | 101 | 1 | *n* |
| (hex ASCII) | 1B | 65 | 01 | *n* |

REMARKS

This command cancels all current vertical tab positions and sets those every *n* lines.

***Note:*** The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.

SEE

Chapter 6

## CONTROLLING THE HORIZONTAL PRINT POSITION

This section described commands that move the print head and restrict its printing range (such as setting margins and tabs).

| | | | |
|---|---|---|---|
| PURPOSE | **Returns print head to the left margin (carriage return).** | | |
| CODE | ⟨CR⟩ | | |
| (decimal ASCII) | 13 | | |
| (hex ASCII) | 0D | | |
| REMARKS | This command returns the print head to the left margin. If DIP switch 2-6 has been set off, then this command will also cause a line feed character to be generated after the carriage return, thereby advancing to the beginning of the next print line automatically. | | |
| SEE | Chapter 6 | | |
| PURPOSE | **Sets the left margin.** | | |
| CODE | ⟨ESC⟩ | "l" | *n* |
| (decimal ASCII) | 27 | 108 | *n* |
| (hex ASCII) | 1B | 6C | *n* |
| REMARKS | This command sets the left margin to *n* characters. Each line will begin in the (*n* + 1)th character position from the left edge. The value of *n* must be between 0 and 255. You can set the left margin manually with the control panel.<br>***Note:*** Changing the print pitch after the left margin has been set does not change the margin — it stays in exactly the same place on the page. | | |
| SEE | Chapter 6 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Sets the right margin.** | | |
| CODE | ⟨ESC⟩ | "Q" | *n* |
| (decimal ASCII) | 27 | 81 | *n* |
| (hex ASCII) | 1B | 51 | *n* |
| REMARKS | This command sets the right margin to *n*, which is the last character position that can be printed in a line. After execution of this command, any attempt to print beyond print position *n* will cause the printer to automatically generate a carriage return and a line feed before printing the remainder of the line. The value of *n* must be between 1 and 255. You can set the right margin manually with the control panel.<br>***Note:*** Changing the print pitch after the right margin has been set does not change the margin — is stays in exactly the same position on the page. | | |
| SEE | Chapter 6 | | |

| | |
|---|---|
| PURPOSE | **Moves the print head to the next horizontal tab position.** |
| CODE | ⟨HT⟩ |
| (decimal ASCII) | 9 |
| (hex ASCII) | 09 |
| REMARKS | This command causes the print head to advance to the next horizontal tab position. The horizontal tab positions are set at power-on to print positions 8, 16, 24, etc. (to the maximum print position). |
| SEE | Chapter 6 |

PURPOSE **Sets horizontal tab positions.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "D" | *n1 n2 n3* ... | 0 |
| (decimal ASCII) | 27 | 68 | *n1 n2 n3* ... | 0 |
| (hex ASCII) | 1B | 44 | *n1 n2 n3* ... | 00 |

REMARKS This command cancels all current horizontal tab positions and sets those defined at print positions *n1, n2, n3,* etc. The maximum number of horizontal tab positions allowed is 32. The ASCII 0 character is used as a command terminator. Each horizontal tab position must be specified in ascending order.

SEE Chapter 6

PURPOSE **Sets horizontal tab positions every *n* characters.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "e" | 0 | *n* |
| (decimal ASCII) | 27 | 101 | 0 | *n* |
| (hex ASCII) | 1B | 65 | 00 | *n* |

REMARKS This command cancels all current horizontal tab positions and sets those every *n* characters.
***Note:*** The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.

SEE Chapter 6

PURPOSE **Moves the print head to an absolute horizontal position.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "$" | *n1* | *n2* |
| (decimal ASCII) | 27 | 36 | *n1* | *n2* |
| (hex ASCII) | 1B | 24 | *n1* | *n2* |

REMARKS This command causes the printer to move the print head to an absolute horizontal position. The position, in inches, is determined by the formula ($n1 + n2 \times 256$)/60. The maximum position is 13.6 inches.

SEE Chapter 6

PURPOSE **Moves the print head to a specified horizontal position.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "\" | *n1* | *n2* |
| (decimal ASCII) | 27 | 92 | *n1* | *n2* |
| (hex ASCII) | 1B | 5C | *n1* | *n2* |

REMARKS This command causes the printer to move the print head to a specified horizontal position. It can move the print head either left or right. The distance, in inches, is determined by the following formulas:

Draft: $(n1 + n2 \times 256)/120$
Letter Quality: $(n1 + n2 \times 256)/180$
Proportional: $(n1 + n2 \times 256)/360$

To move to the left, add 64 to the calculated value of *n2*. The maximum distance is 13.6 inches. The command will be ignored if you try to move to a position outside of the current margins.

SEE Chapter 6

PURPOSE **Adds *n* dot spaces between characters.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | "*space*" | *n* |
| (decimal ASCII) | 27 | 32 | *n* |
| (hex ASCII) | 1B | 20 | *n* |

REMARKS This command increases the space between characters by *n* dots when the DIP switch 2-2 is set off.

SEE Chapter 7

PURPOSE

**Sets the print position to *n* characters.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | "f" | 0 | *n* |
| (decimal ASCII) | 27 | 102 | 0 | *n* |
| (hex ASCII) | 1B | 66 | 00 | *n* |

REMARKS

This command sets the next print position to *n* columns from the left margin. The value of *n* must be between 0 and 127.

***Note:*** The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.

SEE

Chapter 6

## DOWNLOAD CHARACTER COMMANDS

PURPOSE **Defines download characters into RAM.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | “&” | 0 | *n1 n2 m0 m1*<br>*m2 d1 d2 ... dx* |
| (decimal ASCII) | 27 | 38 | 0 | *n1 n2 m0 m1*<br>*m2 d1 d2 ... dx* |
| (hex ASCII) | 1B | 26 | 00 | *n1 n2 m0 m1*<br>*m2 d1 d2 ... dx* |

REMARKS This command is used to set up one or more user-defined characters and store them into RAM for later use. RAM is cleared when the power is turned off. The values of *n1* and *n2* specify the range of positions in RAM that the characters are to occupy. Valid character positions are between 33 and 126. Following *n2* this printer expects character data bytes for each character to be defined. The first byte, *m0*, specifies the left hand space of the download character. The second byte, *m1*, specifies the character width. And the third byte, *m2*, specifies the right hand space of the character. The sum of *m0, m1* and *m2* should be less than 12 for the draft characters and 18 for the LQ characters. *d1* through *dx* determine which dots form the character.

***Note:*** This command is ignored when the DIP switch 2-3 is set on.

SEE Chapter 8

PURPOSE **Copies standard character ROM font into RAM.**

| | | | | | |
|---|---|---|---|---|---|
| CODE | 〈ESC〉 | ":" | 0 | 0 | 0 |
| (decimal ASCII) | 27 | 58 | 0 | 0 | 0 |
| (hex ASCII) | 1B | 3A | 00 | 00 | 00 |

REMARKS This command copies all the standard internal characters to corresponding download character RAM area. This destroys any existing user-defined characters in that range.

***Note:*** This command is ignored when the DIP switch 2-3 is set on.

SEE Chapter 8

PURPOSE **Selects download character set.**

| | | | | |
|---|---|---|---|---|
| CODE | 〈ESC〉 | "%" | 1 | 0 |
| (decimal ASCII) | 27 | 37 | 1 | 0 |
| (hex ASCII) | 1B | 25 | 01 | 00 |

REMARKS This command causes the printer to select the download character set.

***Note:*** The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.

SEE Chapter 8

PURPOSE **Cancels download character set.**

| | | | | |
|---|---|---|---|---|
| CODE | 〈ESC〉 | "%" | 0 | 0 |
| (decimal ASCII) | 27 | 37 | 0 | 0 |
| (hex ASCII) | 1B | 25 | 00 | 00 |

REMARKS This command cancels the download character set and selects the previous character set.

***Note:*** The character "0" (decimal code 48, hexadecimal code 30) can be used instead of the first ASCII 0.

SEE Chapter 8

## DOT GRAPHICS COMMANDS

PURPOSE **Prints normal-density graphics.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | “K” | *n1 n2 m1 m2.....* |
| (decimal ASCII) | 27 | 75 | *n1 n2 m1 m2.....* |
| (hex ASCII) | 1B | 4B | *n1 n2 m1 m2.....* |

REMARKS This command selects 60 dots-per-inch, column-scan, bit-image graphics mode. The values of *n1* and *n2* represent the number of graphics characters to be printed, where the total number of characters = *n2* times 256 + *n1*. The correct number of graphics data bytes (*m1, m2*, etc.) must follow *n2*. The ASCII value of these characters determine which pins are fired for each character.

SEE Chapter 9

PURPOSE **Prints double-density graphics.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | “L” | *n1 n2 m1 m2.....* |
| (decimal ASCII) | 27 | 76 | *n1 n2 m1 m2.....* |
| (hex ASCII) | 1B | 4C | *n1 n2 m1 m2.....* |

REMARKS This command selects 120 dots-per-inch, column-scan, bit-image graphics mode. The values of *n1* and *n2* are the same as in normal-density graphics. The correct number of graphics data bytes (*m1, m2*, etc.) must follow *n2*. The ASCII value of these characters determine which pins are fired for each character.

SEE Chapter 9

PURPOSE **Prints double-density graphics.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | "Y" | *n1 n2 m1 m2.....* |
| (decimal ASCII) | 27 | 89 | *n1 n2 m1 m2.....* |
| (hex ASCII) | 1B | 59 | *n1 n2 m1 m2.....* |

REMARKS Same as ⟨ESC⟩ "L", above.

SEE Chapter 9

PURPOSE **Prints quadruple-density graphics.**

| | | | |
|---|---|---|---|
| CODE | ⟨ESC⟩ | "Z" | *n1 n2 m1 m2.....* |
| (decimal ASCII) | 27 | 90 | *n1 n2 m1 m2.....* |
| (hex ASCII) | 1B | 5A | *n1 n2 m1 m2.....* |

REMARKS This command selects 240 dots-per-inch, column-scan, bit-image graphics mode. The values of *n1* and *n2* are the same as in normal-density graphics. The correct number of graphics data bytes (*m1, m2,* etc.) must follow *n2*. The ASCII value of these characters determine which pins are fired for each character.

SEE Chapter 9

| | |
|---|---|
| PURPOSE | **Selects graphics modes.** |

| | | | |
|---|---|---|---|
| CODE | 〈ESC〉 | "*" | *n0 n1 n2 m1 m2.....* |
| (decimal ASCII) | 27 | 42 | *n0 n1 n2 m1 m2.....* |
| (hex ASCII) | 1B | 2A | *n0 n1 n2 m1 m2.....* |

REMARKS

This command selects one ten possible graphics modes, depending on the value of *n0*. The values of *n1* and *n2* are the same as normal-density graphics mode. The correct number of graphics data bytes (*m1, m2,* etc.) must follow *n2*. The ASCII value of these characters determine which pins are fired for each character. The value of *n0* and its related graphics modes are shown below.

| *n* | Graphics mode |
|---|---|
| 0 | Normal-density (60 dots per inch) |
| 1 | Double-density (120 dots per inch) |
| 2 | Double-density (120 dots per inch) |
| 3 | Quadruple-density (240 dots per inch) |
| 4 | Semi-double density (80 dots per inch) |
| 6 | CRT graphics (90 dots per inch) |
| 32 | 24-pin normal-density (60 dots per inch) |
| 33 | 24-pin double-density (120 dots per inch) |
| 38 | 24-pin CRT graphics (90 dots per inch) |
| 39 | 24-pin triple-density (180 dots per inch) |

SEE Chapter 9

PURPOSE **Redefines the graphics mode.**

| | | | | |
|---|---|---|---|---|
| CODE | ⟨ESC⟩ | “?” | *n0* | *n1* |
| (decimal ASCII) | 27 | 63 | *n0* | *n1* |
| (hex ASCII) | 1B | 3F | *n0* | *n1* |

REMARKS This command redefines one of the 4 alternate graphics commands — ⟨ESC⟩ “K”, ⟨ESC⟩ “L”, ⟨ESC⟩ “Y”, or ⟨ESC⟩ “Z” — as one of the nine graphics density numbers with the ⟨ESC⟩ “*” command, where *n0* is “K”, “L”, “Y”, or “Z” and *n1* is 0, 1, 2, 3, 4, 6, 32, 33, 38, or 39.

SEE Chapter 9

## OTHER COMMANDS

PURPOSE **Sets the value of the eighth data bit to logical 1.**

| | | |
|---|---|---|
| CODE | ⟨ESC⟩ | “⟩” |
| (decimal ASCII) | 27 | 62 |
| (hex ASCII) | 1B | 3E |

REMARKS This command forces the eighth data bit of each subsequent character sent to the printer to logical 1. This code allows users with a 7-bit interface to access those characters whose ASCII code is greater than 127. This code should not be used to transmit printer control codes.

SEE Chapter 7

| | | |
|---|---|---|
| PURPOSE | **Sets the value of the eighth data bit to logical 0.** | |
| CODE | ⟨ESC⟩ | "=" |
| (decimal ASCII) | 27 | 61 |
| (hex ASCII) | 1B | 3D |
| REMARKS | This command forces the eighth data bit of each subsequent character sent to the printer to logical 0. This code should not be used to transmit printer control code. | |
| SEE | Chapter 7 | |

| | | |
|---|---|---|
| PURPOSE | **Accepts the value of the eighth data bit as is.** | |
| CODE | ⟨ESC⟩ | "#" |
| (decimal ASCII) | 27 | 35 |
| (hex ASCII) | 1B | 23 |
| REMARKS | This command cancels either setting of the eighth data bit. The printer will use the value of the eighth data bit that is sent from the computer. This code allows users with a 7-bit interface to resume normal functions after accessing those characters whose ASCII code is greater than 127. | |
| SEE | Chapter 7 | |

| | |
|---|---|
| PURPOSE | **Moves the print head back one print position (backspace).** |
| CODE | ⟨BS⟩ |
| (decimal ASCII) | 8 |
| (hex ASCII) | 08 |
| REMARKS | This command shifts the print head one column to the left. If the print head is at the left margin, the command is ignored. This command can be used to overstrike characters. |
| SEE | Chapter 7 |

PURPOSE **Deletes the last character sent.**

CODE ⟨DEL⟩
(decimal ASCII) 127
(hex ASCII) 7F

REMARKS This command deletes the last character received. This command is ignored if the last character received has already been printed, or if the last character received was all or part of a function code.

SEE Chapter 7

PURPOSE **Cancels line.**

CODE ⟨CAN⟩
(decimal ASCII) 24
(hex ASCII) 18

REMARKS This command deletes the last line in the print buffer at the time the command is used.

SEE Chaper 7

PURPOSE **Sets printer off line.**

CODE ⟨DC3⟩
(decimal ASCII) 19
(hex ASCII) 13

REMARKS This command causes the printer to set itself off line, disregarding all subsequent characters and function codes, with the exception of ⟨DC1⟩, which will return the printer to an on line state. This is not the same as pushing the On Line key. When the On Line lamp is out the printer will not respond to ⟨DC1⟩.

SEE Chapter 7

| | | |
|---|---|---|
| PURPOSE | **Sets printer on line.** | |
| CODE | 〈DC1〉 | |
| (decimal ASCII) | 17 | |
| (hex ASCII) | 11 | |
| REMARKS | This command resets the printer to an on line state, thus allowing it to receive and process all subsequent characters and function codes. This is not the same as pushing the On Line key. When the On Line lamp is out the printer will not respond to 〈DC1〉. | |
| SEE | Chapter 7 | |

| | | |
|---|---|---|
| PURPOSE | **Sounds the printer bell.** | |
| CODE | 〈BEL〉 | |
| (decimal ASCII) | 7 | |
| (hex ASCII) | 07 | |
| REMARKS | This command causes the buzzer to sound for about a quarter of a second. | |
| SEE | Chapter 7 | |

| | | |
|---|---|---|
| PURPOSE | **Disables paper-out detector.** | |
| CODE | 〈ESC〉 | "8" |
| (decimal ASCII) | 27 | 56 |
| (hex ASCII) | 1B | 38 |
| REMARKS | This command causes the printer to disregard the signal sent by the paper-out detector. The paper-out signal normally sounds the printer bell and stops printing until paper is inserted and the printer is reset. DIP switch 2-4 can also set to disable the paper-out detector. | |
| SEE | Chapter 7 | |

| | | | |
|---|---|---|---|
| PURPOSE | **Enables paper-out detector.** | | |
| CODE | 〈ESC〉 | "9" | |
| (decimal ASCII) | 27 | 57 | |
| (hex ASCII) | 1B | 39 | |
| REMARKS | This command restores the function of the paper-out detector. | | |
| SEE | Chapter 7 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Selects uni-directional printing.** | | |
| CODE | 〈ESC〉 | "U" | 1 |
| (decimal ASCII) | 27 | 85 | 1 |
| (hex ASCII) | 1B | 55 | 01 |
| REMARKS | This command causes all subsequent printing to be done in uni-directional printing. Uni-directional printing is useful in printing tables or charts, since it ensures that vertical columns of characters will be in alignment.<br>***Note:*** The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1. | | |
| SEE | Chapter 7 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Cancels uni-directional printing.** | | |
| CODE | 〈ESC〉 | "U" | 0 |
| (decimal ASCII) | 27 | 85 | 0 |
| (hex ASCII) | 1B | 55 | 00 |
| REMARKS | This command cancels uni-directional printing and returns to the standard bi-directional printing, which is considerably faster.<br>***Note:*** The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0. | | |
| SEE | Chapter 7 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Selects one-line uni-directional printing.** | | |
| CODE | ⟨ESC⟩ | "⟨" | |
| (decimal ASCII) | 27 | 60 | |
| (hex ASCII) | 1B | 3C | |
| REMARKS | This command immediately returns the print head to the left margin. The remainder of the line is printed from left to right. Normal (bi-directional) printing resumes following a carriage return. | | |
| SEE | Chapter 7 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Selects double-height expanded printing.** | | |
| CODE | ⟨ESC⟩ | "h" | 1 |
| (decimal ASCII) | 27 | 104 | 1 |
| (hex ASCII) | 1B | 68 | 01 |
| REMARKS | This command causes the printer to print expanded characters with double-height. | | |
| SEE | Chapter 7 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Selects quadruple-height expanded printing.** | | |
| CODE | ⟨ESC⟩ | "h" | 2 |
| (decimal ASCII) | 27 | 104 | 2 |
| (hex ASCII) | 1B | 68 | 02 |
| REMARKS | This command causes the printer to print expanded characters with quadruple-height. | | |
| SEE | Chapter 7 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Cancels double- and quadruple-height expanded printing.** | | |
| CODE | ⟨ESC⟩ | “h” | 0 |
| (decimal ASCII) | 27 | 104 | 0 |
| (hex ASCII) | 1B | 68 | 00 |
| REMARKS | This command cancels double-height and quadruple-height expanded printing, and returns to the previous character size. | | |
| SEE | Chapter 7 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Prints characters in the undefined control code area.** | | |
| CODE | ⟨ESC⟩ | “I” | 1 |
| (decimal ASCII) | 27 | 73 | 1 |
| (hex ASCII) | 1B | 49 | 01 |
| REMARKS | This command causes the printer to print characters in the undefined control code area. This command is ignored when the DIP switch 2-2 is set on.<br>***Note:*** The character “1” (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1. | | |
| SEE | Chapter 8 | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Selects undefined codes as control codes.** | | |
| CODE | ⟨ESC⟩ | “I” | 0 |
| (decimal ASCII) | 27 | 73 | 0 |
| (hex ASCII) | 1B | 49 | 00 |
| REMARKS | This command cancels to print the characters in the undefined control codes and restores them as the control codes.<br>***Note:*** The character “0” (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0. | | |
| SEE | Chapter 8 | | |

PURPOSE **Sets immediate print mode.**

| | | | |
|---|---|---|---|
| CODE | 〈ESC〉 | "i" | 1 |
| (decimal ASCII) | 27 | 105 | 1 |
| (hex ASCII) | 1B | 69 | 01 |

REMARKS This command selects the immediate print mode. In the immediate print mode the print head prints one character at a time, as you send it. The printer also moves the paper up so that you can see the current line and then down to continue printing. This kind of instant feedback can be especially helpful in telecommunications.

***Note:*** The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.

SEE Chapter 7

PURPOSE **Cancels immediate print mode.**

| | | | |
|---|---|---|---|
| CODE | 〈ESC〉 | "i" | 0 |
| (decimal ASCII) | 27 | 105 | 0 |
| (hex ASCII) | 1B | 69 | 00 |

REMARKS This command cancels the immediate print mode and returns the normal print mode.

***Note:*** The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.

SEE Chapter 7

PURPOSE **Repeats the characters.**

| | | | | | | |
|---|---|---|---|---|---|---|
| CODE | 〈ESC〉 | "V" | *n m1 ... mx* | 〈ESC〉 | "V" | 0 |
| (decimal ASCII) | 27 | 86 | *n m1 ... mx* | 27 | 86 | 0 |
| (hex ASCII) | 1B | 56 | *n m1 ... mx* | 1B | 56 | 00 |

REMARKS This command repeats *n* times (*n* is an integer number from 0 to 255) data bytes represented by *m1* ... *mx.*

SEE Chapter 7

PURPOSE **Resets the printer.**

| | | |
|---|---|---|
| CODE | ⟨ESC⟩ | "@" |
| (decimal ASCII) | 27 | 64 |
| (hex ASCII) | 1B | 40 |

REMARKS This command reinitializes the printer. The print buffer is cleared, and the character pitch, character set, line feed pitch, bottom margin, and international character set are all reset to the values defined by their respective DIP switches. The main difference between the ⟨ESC⟩ "@" command and turning the printer off and back on again is that download characters are preserved with this command.

SEE Chapter 7

PURPOSE **Sets 9-pin graphics emulation mode.**

| | | |
|---|---|---|
| CODE | ⟨ESC⟩ | "g" |
| (decimal ASCII) | 27 | 103 |
| (hex ASCII) | 1B | 67 |

REMARKS This command sets the printer to suit for 9-pin printers.

After you've entered into the 9-pin graphics mode, following commands work in the same manner for the 9-pin printers, as shown below:

⟨ESC⟩"A" *n* – Sets or defines line spacing to *n*/72 inch.

⟨ESC⟩"3" *n* – Sets line spacing to *n*/216 inch.

⟨ESC⟩"J" *n* – Sends a one-time paper feed of *n*/216 inch.

⟨ESC⟩"j" *n*– Sends a one-time reverse feed of *n*/216 inch.

In addition, the relationship between the dot graphics data and the pins in the print head will be changed to suit for 9-pin printers.

***Note:*** You must turn off the power switch to cancel this mode.

SEE Chapter 7

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Selects auto feed mode.** | | | | |
| CODE | ⟨ESC⟩ | ⟨EM⟩ | 4 | | |
| (decimal ASCII) | 27 | 25 | 4 | | |
| (hex ASCII) | 1B | 19 | 04 | | |
| REMARKS | This command causes the printer to select the auto sheet feeding mode. This command is ignored when the optional automatic sheet feeder is not mounted on the printer. | | | | |
| SEE | Chapter 7 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Selects auto feed mode.** | | | | |
| CODE | “(” | “(” | “4” | “)” | “)” |
| (decimal ASCII) | 40 | 40 | 52 | 41 | 41 |
| (hex ASCII) | 28 | 28 | 34 | 29 | 29 |
| REMARKS | Same as ⟨ESC⟩ ⟨EM⟩ 4, above. | | | | |
| SEE | Chapter 7 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Cancels auto feed mode.** | | | | |
| CODE | ⟨ESC⟩ | ⟨EM⟩ | 0 | | |
| (decimal ASCII) | 27 | 25 | 0 | | |
| (hex ASCII) | 1B | 19 | 00 | | |
| REMARKS | This command causes the printer to cancel the auto sheet feeding mode. This command is ignored when the optional automatic sheet feeder is not mounted on the printer. | | | | |
| SEE | Chapter 7 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Cancels auto feed mode.** | | | | |
| CODE | “(” | “(” | “0” | “)” | “)” |
| (decimal ASCII) | 40 | 40 | 48 | 41 | 41 |
| (hex ASCII) | 28 | 28 | 30 | 29 | 29 |
| REMARKS | Same as ⟨ESC⟩ ⟨EM⟩ 0, above. | | | | |
| SEE | Chapter 7 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Supplies paper from first bin.** | | | | |
| CODE | ⟨ESC⟩ | ⟨EM⟩ | 1 | | |
| (decimal ASCII) | 27 | 25 | 1 | | |
| (hex ASCII) | 1B | 19 | 01 | | |
| REMARKS | This command causes the printer to supply paper from the first bin. This command is ignored when the optional automatic sheet feeder is not mounted on the printer. | | | | |
| SEE | Chapter 7 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Supplies paper from first bin.** | | | | |
| CODE | “(” | “(” | “1” | “)” | “)” |
| (decimal ASCII) | 40 | 40 | 49 | 41 | 41 |
| (hex ASCII) | 28 | 28 | 31 | 29 | 29 |
| REMARKS | Same as ⟨ESC⟩ ⟨EM⟩ 1, above. | | | | |
| SEE | Chapter 7 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Supplies paper from second bin.** | | | | |
| CODE | ⟨ESC⟩ | ⟨EM⟩ | 2 | | |
| (decimal ASCII) | 27 | 25 | 2 | | |
| (hex ASCII) | 1B | 19 | 02 | | |
| REMARKS | This command causes the printer to supply paper from the second bin. This command is ignored when the optional automatic sheet feeder is not mounted on the printer. | | | | |
| SEE | Chapter 7 | | | | |

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Supplies paper from second bin.** | | | | |
| CODE | “(” | “(” | “2” | “)” | “)” |
| (decimal ASCII) | 40 | 40 | 50 | 41 | 41 |
| (hex ASCII) | 28 | 28 | 32 | 29 | 29 |
| REMARKS | Same as ⟨ESC⟩ ⟨EM⟩ 2, above. | | | | |
| SEE | Chapter 7 | | | | |

| | | | |
|---|---|---|---|
| PURPOSE | **Ejects paper.** | | |
| CODE | ⟨ESC⟩ | ⟨EM⟩ | “R” |
| (decimal ASCII) | 27 | 25 | 82 |
| (hex ASCII) | 1B | 19 | 52 |
| REMARKS | This command causes the printer to eject paper. This command is ignored when the optional automatic sheet feeder is not mounted on the printer. | | |
| SEE | Chapter 7 | | |

| | | | | | |
|---|---|---|---|---|---|
| PURPOSE | **Ejects paper.** | | | | |
| CODE | “(” | “(” | “R” | “)” | “)” |
| (decimal ASCII) | 40 | 40 | 82 | 41 | 41 |
| (hex ASCII) | 28 | 28 | 52 | 29 | 29 |
| REMARKS | Same as ⟨ESC⟩ ⟨EM⟩ “R”, above. | | | | |
| SEE | Chapter 7 | | | | |

# APPENDIX D

# COMMAND SUMMARY IN NUMERIC ORDER

| Control code | Function |
|---|---|
| CHR$(7) | Sounds the printer bell |
| CHR$(8) | Moves the print head back one print position (backspace) |
| CHR$(9) | Moves the print head to the next horizontal tab position |
| CHR$(10) | Advance the paper one line (line feed) |
| CHR$(11) | Advances paper to the next vertical tab position |
| CHR$(12) | Advances the paper to the top of the next page (form feed) |
| CHR$(13) | Returns print head to the left margin (carriage return) |
| CHR$(14) | Sets the printer to expanded print for the remainder of the current line |
| CHR$(15) | Sets the printer to condensed print |
| CHR$(17) | Sets printer on line |
| CHR$(18) | Cancels the condensed print |
| CHR$(19) | Sets printer off line |
| CHR$(20) | Cancels one line expanded print |
| CHR$(24) | Cancels line |
| CHR$(27) | Escape (indicated as 〈ESC〉 below) |
| CHR$(127) | Deletes the last character sent |
| 〈ESC〉 CHR$(14) | Sets the printer to expanded print for the remainder of the current line |
| 〈ESC〉 CHR$(15) | Sets the printer to condensed print |
| 〈ESC〉 CHR$(25) CHR$(0) | Cancels auto feed mode |
| 〈ESC〉 CHR$(25) CHR$(1) | Supplies paper from first bin |
| 〈ESC〉 CHR$(25) CHR$(2) | Supplies paper from second bin |

| | |
|---|---|
| ⟨ESC⟩ CHR$(25) CHR$(4) | Selects auto feed mode |
| ⟨ESC⟩ CHR$(25) "R" | Ejects paper |
| ⟨ESC⟩ CHR$(32) *n* | Adds *n* dot spaces between characters |
| ⟨ESC⟩ "!" *n* | Sets the master print mode |
| ⟨ESC⟩ "#" | Accepts the value of the eighth data bit as is |
| ⟨ESC⟩ "$" *n1 n2* | Moves the print head to an absolute horizontal position |
| ⟨ESC⟩ "%" 0 CHR$(0) | Cancels download character set |
| ⟨ESC⟩ "%" 1 CHR$(0) | Selects download character set |
| ⟨ESC⟩ "&" CHR$(0) *n1 n2 m0 m1 m2 d1 d2... dx* | Defines download characters into RAM |
| ⟨ESC⟩ "*" *n0 n1 n2 m1 m2 ...* | Selects graphics modes |
| ⟨ESC⟩ "−" 0 | Cancels underlining |
| ⟨ESC⟩ "−" 1 | Selects underlining |
| ⟨ESC⟩ "/" *n0* | Selects vertical channel |
| ⟨ESC⟩ "0" | Sets line spacing to 1/8 inch |
| ⟨ESC⟩ "1" | Sets line spacing to 7/60 inch |
| ⟨ESC⟩ "2" | Sets line spacing to 1/6 inch, or uses ⟨ESC⟩"A" definition |
| ⟨ESC⟩ "3" *n* | Sets line spacing to *n*/180 inch |
| ⟨ESC⟩ "4" | Selects optional character set |
| ⟨ESC⟩ "5" | Cancels optional character set |
| ⟨ESC⟩ "6" | Selects character set #2 |
| ⟨ESC⟩ "7" | Selects character set #1 |
| ⟨ESC⟩ "8" | Disables paper-out detector |
| ⟨ESC⟩ "9" | Enables paper-out detector |
| ⟨ESC⟩ ":" CHR$(0) CHR$(0) CHR$(0) | Copies standard ROM font into RAM |
| ⟨ESC⟩ "⟨" | Selects one-line uni-directional printing |
| ⟨ESC⟩ "=" | Sets the value of the eighth data bit to logical 0 |
| ⟨ESC⟩ "⟩" | Sets the value of the eighth data bit to logical 1 |
| ⟨ESC⟩ "?" *n0 n1* | Redefines the graphics mode |
| ⟨ESC⟩ "@" | Resets the printer |
| ⟨ESC⟩ "A" *n* | Sets or defines line spacing to *n*/60 inch |

| | |
|---|---|
| 〈ESC〉 “B” *n1 n2 n3* ... CHR$(0) | Sets vertical tab positions |
| 〈ESC〉 “C” CHR$(0) *n* | Sets page length to *n* inches |
| 〈ESC〉 “C” *n* | Sets page length to *n* lines |
| 〈ESC〉 “D” *n1 n2 n3* ... CHR$(0) | Sets horizonal tab positions |
| 〈ESC〉 “E” | Selects boldface printing |
| 〈ESC〉 “F” | Cancels boldface printing |
| 〈ESC〉 “G” | Selects boldface printing |
| 〈ESC〉 “H” | Cancels boldface printing |
| 〈ESC〉 “I” 0 | Selects undefined codes as control codes |
| 〈ESC〉 “I” 1 | Prints characters in the undefined control code area |
| 〈ESC〉 “J” *n* | Sends a one-time paper feed of *n*/180 inch |
| 〈ESC〉 “K” *n1 n2 m1 m2* ... | Prints normal-density graphics |
| 〈ESC〉 “L” *n1 n2 m1 m2* ... | Prints double-density graphics |
| 〈ESC〉 “M” | Sets the print pitch to elite |
| 〈ESC〉 “N” *n* | Sets the bottom margin |
| 〈ESC〉 “O” | Cancels the bottom margin |
| 〈ESC〉 “P” | Sets the print pitch to pica |
| 〈ESC〉 “Q” *n* | Sets the right margin |
| 〈ESC〉 “R” *n* | Selects an international character set |
| 〈ESC〉 “S” 0 | Selects superscripts |
| 〈ESC〉 “S” 1 | Selects subscripts |
| 〈ESC〉 “T” | Cancels a superscript or subscript |
| 〈ESC〉 “U” 0 | Cancels uni-directional printing |
| 〈ESC〉 “U” 1 | Selects uni-directional printing |
| 〈ESC〉 “V” *n m1* ..... *mx* 〈ESC〉“V”CHR$(0) | Repeats the characters |
| 〈ESC〉 “W” 0 | Cancels the expanded print |
| 〈ESC〉 “W” 1 | Sets the printer to expanded print |
| 〈ESC〉 “Y” *n1 n2 m1 m2* ... | Prints double-density graphics |
| 〈ESC〉 “Z” *n1 n2 m1 m2*... | Prints quadruple-density graphics |
| 〈ESC〉 “\” *n1 n2* | Moves the print head to a specified horizontal position |
| 〈ESC〉 “_” 0 | Cancels overlining |
| 〈ESC〉 “_” 1 | Selects overlining |
| 〈ESC〉 “b” *n0 n1 n2 n3* ... CHR$(0) | Sets vertical tab positions in a channel |

| | |
|---|---|
| ⟨ESC⟩ “e” 0 *n* | Sets horizontal tab positions every *n* characters |
| ⟨ESC⟩ “e” 1 *n* | Sets vertical tab positions every *n* lines |
| ⟨ESC⟩ “f” 0 *n* | Sets the print position to *n* characters |
| ⟨ESC⟩ “f” 1 *n* | Sets print position to *n* lines |
| ⟨ESC⟩ “g” | Sets 9-pin graphics emulation mode |
| ⟨ESC⟩ “h” CHR$(0) | Cancels double- and quadruple-height expanded printing |
| ⟨ESC⟩ “h” CHR$(1) | Selects double-height expanded printing |
| ⟨ESC⟩ “h” CHR$(2) | Selects quadruple-height expanded printing |
| ⟨ESC⟩ “i” 0 | Cancels immediate print mode |
| ⟨ESC⟩ “i” 1 | Sets immediate print mode |
| ⟨ESC⟩ “j” *n* | Sends a one-time reverse feed of *n*/180 inch |
| ⟨ESC⟩ “k” *n* | Selects a character set |
| ⟨ESC⟩ “l” *n* | Sets the left margin |
| ⟨ESC⟩ “p” 0 | Cancels proportional print |
| ⟨ESC⟩ “p” 1 | Sets the printer to proportional print |
| ⟨ESC⟩ “x” 0 | Cancels the LQ characters |
| ⟨ESC⟩ “x” 1 | Selects the LQ characters |
| “((0))” | Cancels auto feed mode |
| “((1))” | Supplies paper from first bin |
| “((2))” | Supplies paper from second bin |
| “((4))” | Selects auto feed mode |
| “((R))” | Ejects paper |

APPENDIX E

# TECHNICAL SPECIFICATIONS

***Printing***

| | |
|---|---|
| Printing method | Serial impact dot matrix |
| Printing speed | 300 characters per second (in Draft elite)<br>250 characters per second (in Draft pica)<br>100 characters per second (in LQ elite)<br>83 characters per second (in LQ pica) |
| Print buffer | 16 KB (Expandable to 32K bytes with optional buffer board) |
| Paper feed | 3.3 inches/second (in case of form feeding)<br>Tractor and Friction feed |
| Printing direction | Bi-directional, logic seeking<br>Uni-directional in dot graphics modes |
| Character set | |
| Draft characters | 96 standard ASCII characters<br>33 international characters [11 sets]<br>81 IBM special characters |
| LQ characters | 96 standard ASCII characters<br>33 international characters [11 sets]<br>81 IBM special characters |
| Other characters | 52 IBM block graphics characters<br>128 super and subscripts<br>128 downloadable characters |

| | |
|---|---|
| Character matrix | 24 × 15 dots, LQ pica characters |
| | 24 × 13 dots, LQ elite characters |
| | 24 × 9 dots, Draft characters |
| | 12 × 8 dots, Super and subscripts |
| | 30 × 18 dots, IBM block graphics |
| | 8 or 24 × 810 dots, Normal-density graphics |
| | 8 × 1080 dots, Semi-double density graphics |
| | 8 or 24 × 1215 dots, CRT graphics |
| | 8 or 24 × 1620 dots, Double-density graphics |
| | 24 × 2430 dots, Triple-density graphics |
| | 8 × 3240 dots, Quadruple-density graphics |
| Line spacing | 1/6 or 1/8 inch standard |
| | *n*/60 or *n*/180 inch programmable |
| Column width | 136, normal pica |
| | 163, normal elite |
| | 222, condensed pica |
| | 244, condensed elite |
| | 68, expanded pica |
| | 81, expanded elite |
| | 111, expanded condensed pica |
| | 122, expanded condensed elite |
| | and proportional spacing |
| Special features | Automatic single sheet insertion |
| | Prestige Letter Quality printing |
| | Short form tear-off |
| | Easy access format switches |
| | Self-test and hex dump |
| | Downloadable characters |
| | 7 or 8 bit selectable interface |
| | Ultra hi-resolution bit image graphics |
| | Vertical and horizontal tabs |
| | Skip over perforation |
| | 15.5″ carriage |
| | Automatic sheet feeder (option) |
| | Various LQ character cartridges (option) |

***Paper***

| | |
|---|---|
| Single sheets | 5.5 – 14.5 inches, wide<br>0.07 – 0.10 mm, thickness |
| Sprocket-feed paper | 4 – 15.5 inches, wide<br>0.07 – 0.10 mm, one-part form thickness<br>Max 0.2 mm, 3-part form thickness |

***Printer***

| | |
|---|---|
| Dimensions | Height 121 mm (4.7 inches)<br>Width 580 mm (22.8 inches)<br>Depth 383 mm (15.1 inches) |
| Weight | 14.8 Kg (32.6 pounds) |
| Power | 120 VAC ± 10%, 60Hz. 275W<br>220 VAC ± 10%, 50/60Hz. 275W<br>240 VAC ± 10%, 50/60Hz. 275W |
| Environment | Temperature: 5 to 35℃ (40 to 95°F)<br>Humidity: 10 to 80%, non condensing |
| Ribbon | Black cloth ribbon in special cartridge<br>Ribbon life: 4.5 × $10^6$ draft characters |
| Print head life | 2 × $10^8$ strokes per wire |

***Parallel interface***

| | |
|---|---|
| Interface | Centronic-compatible, 7 or 8 bit |
| Synchronization | By external supplied Strobe pulses |
| Handshaking | By ACK or BUSY signals |
| Logic level | TTL |
| Connector | 57-30360 Amphenol |

***Serial interface (option)***

| | |
|---|---|
| Interface | Asynchronous RS-232C/20 mA current loop |
| Bit rate | 150, 300, 600, 1200, 2400, 4800, 9600, 19200 baud |
| Word length | 1 start bit<br>7 or 8 data bits<br>Odd, even or no parity<br>1 or 2 stop bits |
| Handshaking | Serial BUSY, 1 byte mode<br>Serial BUSY, 1 block mode<br>ACK mode<br>XON/XOFF mode |

# MEMO

# APPENDIX F

# THE PARALLEL INTERFACE

This printer has a parallel interface to communicate with the computer that it is connected to. The operating specifications of the parallel interface are as follows:

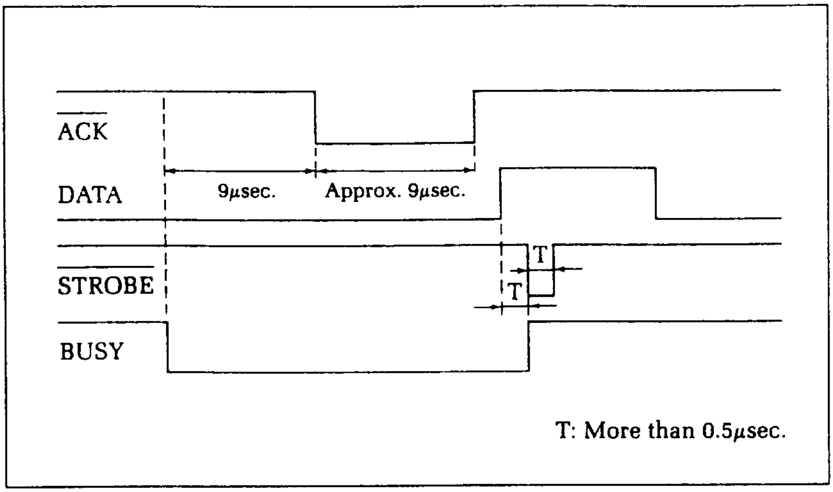
| | |
|---|---|
| Data transfer rate: | 1,000 to 6,000 characters per second |
| Synchronization: | Via externally supplied $\overline{\text{STROBE}}$ pulses |
| Handshaking: | $\overline{\text{ACK}}$ and BUSY signals |
| Logic level: | Compatible with TTL level |

The parallel interface connects to the computer by a 36 pin connector on the back of the printer. This connector mates with an Amphenol 57-30360 connector. The functions of the various pins are summarized in Table F-1.

## ■ Functions of the Connector Signals

Communications between the computer and the printer use many of the pins of the connector. To understand how the system of communications works we need to look at the functions of the various signals carried by the pins of the interface connector.

Pin 1 carries the $\overline{\text{STROBE}}$ pulse signal from the computer to the printer. This signal is normally held high by the computer. When the computer has data ready for the printer it sets this signal to a low value for at least 0.5 microseconds. When the printer sees this pulse on the strobe pin, it reads the data that the computer supplies on pins 2 through 9. Each of these lines carries one bit of information. A logical "1" is represented by a high signal level, and a logical "0" is represented by a low signal level. The computer must maintain these signals for a period beginning at least 0.5 microseconds before the strobe pulse starts and continuing for at least 0.5 microseconds after the strobe pulse ends.



T: More than 0.5μsec.

**Figure F-1.** The interface timing diagram.

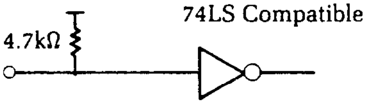
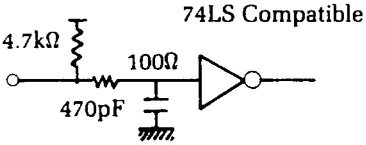
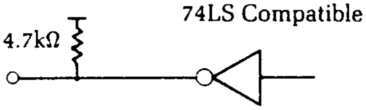
| Signal Name | Circuit Example |
|---|---|
| DATA 1 - DATA 8<br>(To Printer) |  |
| $\overline{\text{STROBE}}$<br>(To Printer) |  |
| BUSY, $\overline{\text{ACK}}$<br>(From Printer) |  |

**Figure F-2.** Typical interface circuit.

When the printer has successfully received the byte of data from the computer it sets pin 10 low for approximately 9 microseconds. This signal acknowledges the receipt of the data and so is called the $\overline{\text{ACK}}$ (for "acknowledge") signal.

Pin 11 reports when the printer is not able to receive data. The signal is called BUSY. When this signal is high, the printer cannot receive data. This signal will be high during data transfer, when the printer is off-line and when an error condition exists.

The printer will report that it has run out of paper by making the PAPER OUT signal on pin 12 high. This pin can be held low by turning DIP switch 2-4 off. When the printer is in the on-line state pin 13 is held high. This signal (SELECTED) tells the computer that the printer is ready to receive data.

Pins 14, 15, 34 and 35 are not used, while pins 16, 17, 19—30

**Table F-1**
**Parallel interface pin functions**

| Pin No. | Signal Name | Direction | Function |
|---|---|---|---|
| 1 | $\overline{\text{STROBE}}$ | IN | Signals when data is ready to be read. Signal goes from HIGH to LOW (for at least 0.5 microseconds) when data is available. |
| 2 | DATA1 | IN | These signals provide the information of the first to eighth bits of parallel data. Each signal is at HIGH level for a logical 1 and at a LOW level for a logical 0. |
| 3 | DATA2 | IN | |
| 4 | DATA3 | IN | |
| 5 | DATA4 | IN | |
| 6 | DATA5 | IN | |
| 7 | DATA6 | IN | |
| 8 | DATA7 | IN | |
| 9 | DATA8 | IN | |
| 10 | $\overline{\text{ACK}}$ | OUT | A 9 microsecond LOW pulse acknowledges receipt of data. |
| 11 | BUSY | OUT | When this signal goes LOW the printer is ready to accept data. |
| 12 | PAPER OUT | OUT | This signal is normally LOW. It will go HIGH if the printer runs out of paper. This signal can be held LOW permanently by turning DIP switch 2-4 off. |
| 13 | SELECTED | OUT | This signal is HIGH when the printer is on-line. |
| 14-15 | N/C | | Unused |
| 16 | SIGNAL GND | | Signal ground. |
| 17 | CHASSIS GND | | Printer's chassis ground, isolated from logic ground. |
| 18 | + 5VDC | OUT | External supply of + 5VDC. |
| 19-30 | GND | | Twisted pair return signal ground level. |
| 31 | $\overline{\text{RESET}}$ | IN | When this signal goes LOW the printer is reset to its power-on condition. |
| 32 | $\overline{\text{ERROR}}$ | OUT | This signal is normally HIGH. This signal goes LOW to signal that the printer cannot print due to an error condition. |
| 33 | EXT GND | | External ground. |
| 34,35 | N/C | | Unused. |
| 36 | $\overline{\text{SELECT IN}}$ | | Data entry to the printer is possible only when this level is LOW. |

and 33 are grounded. Pin 18 is connected to the + 5VDC supply in the printer.

Pin 31 can be used to reset the printer. If this signal ($\overline{\text{RESET}}$) goes low the printer will reinitialize. Pin 32 is used to report error conditions in the printer. This signal ($\overline{\text{ERROR}}$) is high during normal operation and goes low to report that the printer cannot print due to an error condition.

# APPENDIX G
# SERIAL INTERFACE SPECIFICATIONS

This printer provides a very flexible RS232C serial interface as an option. It can communicate at rates from 150 to 19,200 baud (bits per second) and supports four different kinds of *handshaking*. This interface can also function as a 20mA current loop interface. The operating specifications of the interface are as follows:

| | |
|---|---|
| Data transfer rate: | 150-19200 |
| Word length: | 1 start bit<br>7 or 8 data bits<br>Odd, even or no parity<br>1 or 2 stop bits |
| Signal levels: | Mark or logical 1, −3 to −15 volts or current ON<br>Space or logical 0, +3 to +15 volts or current OFF |
| Handshaking: | Serial BUSY, 1 byte mode<br>Serial BUSY, 1 block mode<br>ACK mode<br>XON/XOFF mode |

***Note:*** 19200 baud can be used only with an RS232C interface; it cannot be used with a 20mA current loop interface.

The optional board has a DB-25 female connector to connect to a computer. The functions of the pins are summarized in Table G-1.

## CONFIGURING THE SERIAL INTERFACE

DIP switch on the serial interface board controls the configuration of the serial interface. Table G-2 describes the functions of the individual switches in DIP switch.

## Table G-1
## Serial interface pin functions

| Pin No. | Signal Name | Direction | Function |
|---|---|---|---|
| 1 | GND | — | Printer's chassis ground. |
| 2 | TXD | OUT | This pin carries data from the printer. |
| 3 | RXD | IN | This pin carries data to the printer. |
| 4 | RTS | OUT | This is ON when the printer is ready to receive data. |
| 5 | CTS | IN | This pin is ON when the computer is ready to send data. |
| 6 | DSR | IN | This pin is ON when the computer is ready to send data. This printer does not check this pin. |
| 7 | GND | — | Signal ground. |
| 8 | DCD | IN | This pin is ON when the computer is ready to send data. This printer does not check this pin. |
| 9 | TTY TXDR | — | This pin is the return path for data transmitted from the printer on the 20mA current loop. |
| 10 | TTY TXD | OUT | This pin carries data from the printer on the 20mA current loop. |
| 11 | RCH | OUT | This is the signal line for the serial busy protocols. This pin goes OFF when printer's buffer fills, and ON when the printer is ready to receive data. In the busy protocols this line carries the same signal as pin 20. |
| 12 | N/C | | Unused. |
| 13 | GND | — | Signal ground. |
| 14-16 | N/C | | Unused. |
| 17 | TTY TXDR | — | This pin is the return path for data transmitted from the printer on the 20mA current loop. |
| 18 | TTY RXDR | — | This pin is the return path for data transmitted to the printer on the 20mA current loop. |
| 19 | TTY RXD | IN | This pin carries data to the printer on the 20mA current loop. |
| 20 | DTR | OUT | The printer turns this pin ON when it is ready to receive data. |
| 21-22 | N/C | | Unused. |
| 23 | TTY RXDR | — | This pin is the return path for data transmitted to the printer on the 20mA current loop. |
| 24 | TTY TXD | OUT | This pin carries data from the printer on the 20mA current loop. |
| 25 | TTY RXD | IN | This pin carries data to the printer on the 20mA current loop. |

**Table G-2**
**DIP switch on serial board**

| Switch | ON | OFF |
|---|---|---|
| 1 | 7 data bits | 8 data bits |
| 2 | Parity checked | No parity |
| 3 | Handshaking protocols—see Table G-3 | |
| 4 | | |
| 5 | Odd parity | Even parity |
| 6 | Data transfer rate—see Table G-4 | |
| 7 | | |
| 8 | | |

**Table G-3**
**Handshaking protocols**

| Protocol | Switch 3 | Switch 4 |
|---|---|---|
| Serial busy, 1 byte mode | OFF | OFF |
| Serial busy, 1 block mode | ON | OFF |
| ACK mode | OFF | ON |
| XON/XOFF mode | ON | ON |

**Table G-4**
**Data transfer rates**

| Baud rate | Switch 6 | Switch 7 | Switch 8 |
|---|---|---|---|
| 150 | OFF | OFF | OFF |
| 300 | OFF | OFF | ON |
| 600 | OFF | ON | OFF |
| 1200 | OFF | ON | ON |
| 2400 | ON | OFF | OFF |
| 4800 | ON | OFF | ON |
| 9600 | ON | ON | OFF |
| 19200 | ON | ON | ON |

## THE SERIAL PROTOCOLS

This printer has four serial protocols selected by DIP switches 4 and 5. Figure G-1 shows a typical byte of serial data and Figure G-2 shows timing charts for the 4 protocols.

### ■ Serial busy protocols

In the serial busy protocols, this printer uses DTR (pin 20) and RCH (pin 11) to signal to the computer when it is able to accept data. These two pins go ON when the printer is ready to accept data. In the 1 byte mode they go OFF after each character is received. In the 1 block mode they only go OFF when the printer's buffer approaches capacity. In both cases they will stay OFF if the buffer is too full to accept more data.

### ■ XON/XOFF protocol

The XON/XOFF protocol uses the ASCII characters ⟨DC1⟩ and ⟨DC3⟩ (sometimes called XON and XOFF, respectively) to communicate with the computer. When the printer's buffer approaches capacity this printer will send a DC3 (ASCII 19) on TXD (pin 2) to tell the computer that it must stop sending data. When the printer is able to receive more data it sends a DC1 (ASCII 17) on TXD. The computer can then send more data until the printer sends another DC3.

### ■ ACK protocol

In the ACK protocol, this printer sends an ACK (ASCII 6) on TXD (pin 2) each time that it is prepared to receive a byte of data.

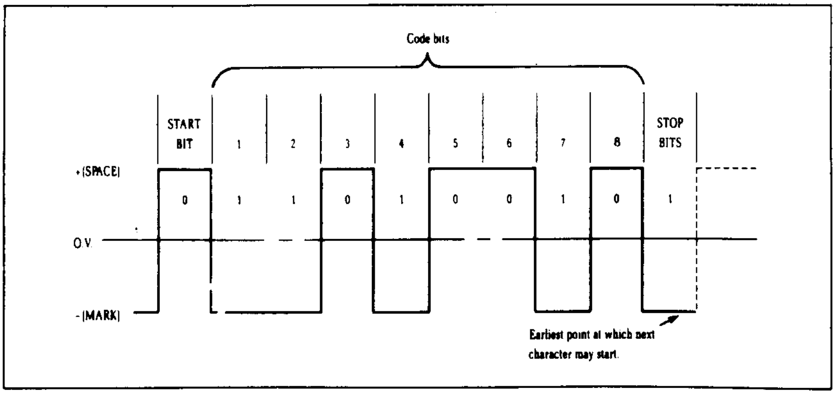


**Figure G-1.** Typical data byte on the serial interface.

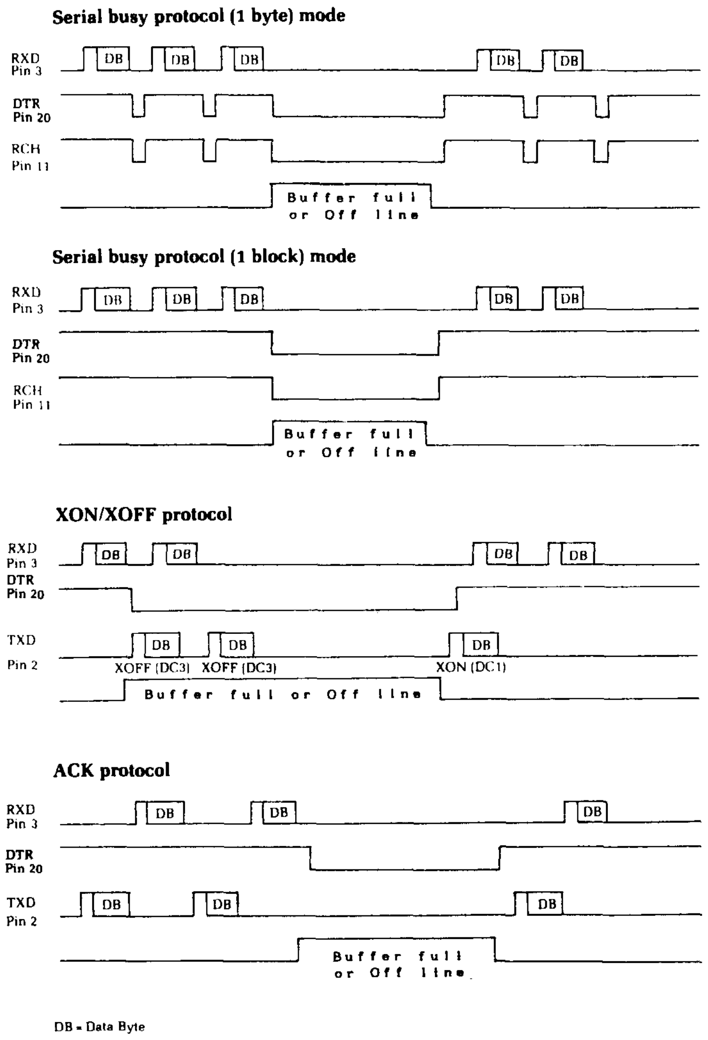


**Figure G-2.** Serial protocol timing charts.

# MEMO

# APPENDIX H

# CONNECTING WITH COMPUTER

In this appendix, we'll show you how to connect with various computers.

If you cannot find out the name of your computer, your printer dealer will give you advice on connecting this printer to your computer.

## CONNECTING WITH IBM-PC AND COMPAQ

Both the IBM Personal Computer and the Compaq computer function the same when connected to this printer. We will discuss the IBM-PC, knowing that all we say works just as well as for the Compaq.

You only need a cable to connect this printer to your IBM-PC. Your printer dealer can furnish this cable, or you can use a standard IBM-PC parallel printer cable for the parallel interface.

**Table H-1**
**IBM-PC parallel cable**

| Printer | | IBM-PC Parallel | |
|---|---|---|---|
| **Pin No.** | **Function** | **Pin No.** | **Function** |
| 1 | STROBE | 1 | STROBE |
| 2 | D1 | 2 | D0 |
| 3 | D2 | 3 | D1 |
| 4 | D3 | 4 | D2 |
| 5 | D4 | 5 | D3 |
| 6 | D5 | 6 | D4 |
| 7 | D6 | 7 | D5 |
| 8 | D7 | 8 | D6 |
| 9 | D8 | 9 | D7 |
| 10 | ACK | 10 | ACK |
| 11 | BUSY | 11 | BUSY |
| 12 | PAPER END | 12 | PAPER END |
| 13 | SELECTED | 13 | SELECT |
| 16 | GROUND | 18-25 | GROUND |
| 31 | RESET | 16 | RESET |
| 32 | ERROR | 15 | ERROR |

### ■ BASIC programming

When you start writing your own programs there are several things you should know.

IBM BASIC defaults to a printer width of 80. This means that it will automatically insert a carriage return and line feed after every 80 characters. If you want to print lines longer than 80 characters you will need to change the width of the printer. If you set the printer width to 255, then the IBM will *never* insert a line feed and carriage return, unless you start a new line. (This is what you want usually.) To set the width of the printer to 255, use this statement:

```
100 WIDTH "LPT1:", 255
```

IBM BASIC has one other little trick that will mess up your graphics if you let it. IBM BASIC is very insistent about adding a line feed to a carriage return. This is fine if you are printing text, but if an ASCII 13 pops up in the middle of your graphics printout, IBM BASIC will *still* add a line feed to it. This will put strange things in the middle of your graphics, and leave you with extra characters at the end of your line.

There is an easy way to avoid this problem. You just open the printer as a random file. The following program shows how this is done.

```
10 OPEN "LPT1:" AS #1          ' RANDOM ACCESS
20 WIDTH #1, 255               ' SET WIDTH TO 255
30 PRINT #1, "TESTING"         ' PRINT A LINE
40 PRINT #1, CHR$(10)          ' ADD YOUR OWN LF
```

### ■ Listing programs

To list programs on this printer, make sure the program is in the IBM's memory and use the LLIST command. This directs the listing to the printer instead of the screen.

## CONNECTING WITH APPLE II COMPUTERS

Apple II computers require an interface board (mounted inside the Apple II) and a cable to run this printer. We recommend that you use the **grafstar**™ interface for the Apple II, II + , and IIe. It comes complete with a cable and is easily installed. A unique feature of the **grafstar**™ makes it possible to do some fancy dot graphics programming.

You can, of course, use many of the available parallel interface boards for the Apple II, and an appropriate cable.

**Table H-2**
**Apple parallel cable**

| **Printer** | | **Apple Board** | |
|---|---|---|---|
| **Pin No.** | **Function** | **Pin No.** | **Function** |
| 25 | SIG GND | 1 | SIG GND |
| 26 | SIG GND | 2 | SIG GND |
| 27 | SIG GND | 3 | SIG GND |
| 1 | STROBE | 4 | STROBE |
| 28 | SIG GND | 5 | N/C |
| 2 | DATA1 | 6 | DATA1 |
| 3 | DATA2 | 7 | DATA2 |
| 4 | DATA3 | 8 | DATA3 |
| 5 | DATA4 | 9 | DATA4 |
| 6 | DATA5 | 10 | DATA5 |
| 7 | DATA6 | 11 | DATA6 |
| 8 | DATA7 | 12 | DATA7 |
| 9 | DATA8 | 13 | DATA8 |
| 10 | ACK | 14 | ACK |
| 29 | SIG GND | 15 | SIG GND |

## ■ Applesoft BASIC

The Apple II computer, using Applesoft BASIC, does not have different types of PRINT statements for the screen and printer. You must add commands to your programs that direct the output of the PRINT statements to the printer. To direct output to the printer (with the interface board in slot # 1) you must use the PR # 1 command. Depending on the version of Applesoft BASIC that you are using this command can take various forms. It is usually one of the following:

```
10 PR#1
or
10 PRINT "<Ctrl-D> PR#1"
or
10 PRINT CHR$(4) "PR#1"
```

To return output to the screen, the command is PR # 0, in the same form that works for PR # 1.

To allow line length longer than the Apple II usually uses you must add the following statement to your programs:

```
20 PRINT CHR$(9) "255N"
```

This allows lines of any length to be sent to the printer and is especially important for dot graphics. (The number 255 in the BASIC statement above could be replaced by any number from 0 to 255 and would set the line length to that value.)

Two codes are particular problem on the Apple II: CHR$(7) and CHR$(9). The computer will not send these codes to this printer. Try to avoid using these in dot graphics programs.

The Apple II computer uses CHR$(9) as a printer initialization code. It won't send it on to the printer. There is a way to bypass this problem, however. You can change the printer initialization code to a value other than CHR$(9) like this:

```
PR#1
PRINT CHR$(9); CHR$(I)
```

This makes CHR$(1) the printer initialization code (and transfers the problems to *that* code) and allows you to use this printer's tabs.

There is one more way to sneak problem codes past the Apple II's operating system and that's to poke the codes directly to the output port. To send ASCII code 9, for example, you could do this:

```
100 N = 9
110 IF PEEK(49601)>127 THEN 110
120 POKE 49296,N
```

Line 110 checks the printer's status, and when it's okay, line 120 pokes the code to the printer.

### ■ Listing programs

To make a listing of your BASIC programs on this printer from your Apple II computer you must take the following steps:

1. Be sure that the program that you wish to list is in the memory of the Apple II.
2. Direct the output to the printer by typing PR # 1.
3. Type LIST to start the listing.

4. When the listing is finished, type PR # 0 to redirect the output to the screen.

## CONNECTING WITH TRS-80 COMPUTERS

All that's required to connect this printer to your TRS-80 is a cable. It is available at your printer dealer.

**Table H-3**
**TRS-80 Model I parallel cable**

| Printer | | | TRS-80 Model I | |
|---|---|---|---|---|
| **Pin No.** | **Function** | | **Pin No.** | **Function** |
| 1 | STROBE | ——— | 1 | STROBE |
| 2 | D1 | ——— | 3 | D1 |
| 3 | D2 | ——— | 5 | D2 |
| 4 | D3 | ——— | 7 | D3 |
| 5 | D4 | ——— | 9 | D4 |
| 6 | D5 | ——— | 11 | D5 |
| 7 | D6 | ——— | 13 | D6 |
| 8 | D7 | ——— | 15 | D7 |
| 9 | D8 | ——— | 17 | D8 |
| 11 | BUSY | ——— | 21 | READY |

**Table H-4**
**TRS-80 Model II parallel cable**

| Printer | | | TRS-80 Model II | |
|---|---|---|---|---|
| **Pin No.** | **Function** | | **Pin No.** | **Function** |
| 1 | STROBE | ——— | 1 | STROBE |
| 2 | D1 | ——— | 3 | D1 |
| 3 | D2 | ——— | 5 | D2 |
| 4 | D3 | ——— | 7 | D3 |
| 5 | D4 | ——— | 9 | D4 |
| 6 | D5 | ——— | 11 | D5 |
| 7 | D6 | ——— | 13 | D6 |
| 8 | D7 | ——— | 15 | D7 |
| 9 | D8 | ——— | 17 | D8 |
| 10 | ACK | ——— | 19 | ACK |
| 11 | BUSY | ——— | 21 | BUSY |

### ■ TRS-80 BASIC

You may have to initialize your Model II to direct LPRINT statements to the printer. Use the SYSTEM "FORMS" command to do it.

TRS-80 uses another version of Microsoft BASIC. TRS-80 does have a few unique "problem codes". They are 0, 10, 11, and 12. None of these are passed properly to the printer.

You can bypass the TRS-80's BASIC and send these codes directly to the printer with the following short routine. The variable N must be set equal to the code that you wish to pass (in our example it's 0).

```
90 N = 0
100 IF PEEK(14312)<>63 THEN 100
110 POKE 14312,N
```

Or you can use this special printer driver that will solve all your problems. Just run this program first, and then any codes sent by a BASIC program will be sent directly to the printer. This program is for the TRS-80 Model III.

```
5 REM   DRIVER FOR TRS-80 III
10 AD=16571
20 FOR I=0 TO 14
30 READ A: POKE AD+1,A
40 NEXT I
50 POKE 16422,187
60 POKE 16423,64
70 DATA 33,232,55,203,126,32,252,33,17,0,57,126,
  211,251,201
80 END
```

And here is a version for the TRS-80 Model I.

```
5 REM   DRIVER FOR TRS-80 I
10 AD=16571
20 FOR I=0 TO 15
30 READ A: POKE AD+1,A
40 NEXT I
50 POKE 16422,187
60 POKE 16423,64
70 DATA 33,232,55,203,126,32,252,33,17,0,57,126,50,
  232,55,201
80 END
```

■ Listing programs

To list a BASIC program that is in your TRS-80's memory on this printer, type LLIST. This directs the listing to the printer instead of the screen.

## CONNECTING WITH KAYPRO, OSBORNE, AND OTHER CP/M COMPUTERS

All that you need to connect this printer to an Osborne 1 or Kaypro computer is a cable. Your printer dealer can provide the cable that you need.

**Table H-5**
**Kaypro parallel cable**

| Printer | | Kaypro | |
|---|---|---|---|
| **Pin No.** | **Function** | **Pin No.** | **Function** |
| 1 | STROBE | 1 | STROBE |
| 2 | DATA1 | 2 | DATA1 |
| 3 | DATA2 | 3 | DATA2 |
| 4 | DATA3 | 4 | DATA3 |
| 5 | DATA4 | 5 | DATA4 |
| 6 | DATA5 | 6 | DATA5 |
| 7 | DATA6 | 7 | DATA6 |
| 8 | DATA7 | 8 | DATA7 |
| 9 | DATA8 | 9 | DATA8 |
| 11 | BUSY | 11 | BUSY |
| 16 | SIG GND | 16 | SIG GND |

**Table H-6**
**Osborne 1 parallel cable**

| Printer | | Osborne 1 | |
|---|---|---|---|
| **Pin No.** | **Function** | **Pin No.** | **Function** |
| 2 | DATA1 | 1 | DATA0 |
| 6 | DATA5 | 2 | DATA4 |
| 3 | DATA2 | 3 | DATA1 |
| 7 | DATA6 | 4 | DATA5 |
| 4 | DATA3 | 5 | DATA2 |
| 8 | DATA7 | 6 | DATA6 |
| 5 | DATA4 | 7 | DATA3 |
| 9 | DATA8 | 8 | DATA7 |
| 1 | STROBE | 11 | STROBE |
| 11 | BUSY | 15 | BUSY |
| 16 | SIG GND | 16 | SIG GND |

### ■ Using MBASIC

Many CP/M computers use Microsoft BASIC (called MBASIC). MBASIC is a very close relative of the IBM-Microsoft BASIC. The only difference is that MBASIC “interprets" CHR$(9) and substitutes a group of spaces to simulate a tab. You can send a horizontal tab to this printer by using CHR$(137) instead of CHR$(9).

Some versions of Microsoft BASIC will add a carriage return and line feed at the end of every 80 (or sometimes 132) characters. To print lines longer than 80 (or 132) characters (as when doing dot graphics) you must define a wider printer width. The following statement will prevent the computer from inserting unwanted codes.

```
10 WIDTH LPRINT 255
```

■ Listing programs

Microsoft BASIC uses the "L" prefix on several commands to direct them to the printer. To list programs on the printer, just type LLIST. To direct program output to the printer, use LPRINT in place of PRINT.