

APPLE IIc
INTERFACE CARTRIDGE
FOR NL-10

USERS MANUAL

APPLE IIc
INTERFACE CARTRIDGE
FOR NL-10
USER'S MANUAL

**Federal Communications Commission
Radio Frequency Interference
Statement**

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

- Reorient the receiving antenna
- Relocate the computer with respect to the receiver
- Move the computer away from the receiver
- Plug the computer into a different outlet so that computer and receiver are on different branch circuits.

If necessary, the user should consult the dealer or an experienced radio/television technician for additional suggestions. The user may find the following booklet, prepared by the Federal Communications Commission helpful: "How to Identify and Resolve Radio-TV Interference Problems." This booklet is available from the U.S. Government Printing Office, Washington, D.C., 20402, Stock No. 004-000-00345-4.

For compliance with Federal Noise Interference Standard, this equipment requires a shielded cable.

This statement will be applied only for the printers marketed in U.S.A.

Self Declaration

Radio interferences regarding this equipment has been eliminated according to Vfg 1046/1984 announced by the DBP.

DBP has been informed about the introduction of this special equipment and has been conceded the right to examine the whole series.

It is upon the responsibility of the user to assume that his own assembled system is in accordance with the technical regulations under Vfg 1046/1984.

To observe FTZ-regulations it is necessary, to establish all connections to the printer with shielded cable.

The equipment may only be opened by qualified service representatives.

This statement will be applied only for the printers marketed in West Germany.

Trademark Acknowledgement

NL-10: Star Micronics Co., Ltd.

Apple, Apple IIc, Applesoft: Apple computer Inc.

Commodore C-64: Commodore Business Machines, Inc.

NOTICE

- All rights reserved. Reproduction of any part of this manual in any form whatsoever, without STAR's express permission is forbidden.
- The contents of this manual are subject to change without notice.
- All efforts have been made to ensure the accuracy of the contents of this manual at the time of going to press. However, should any errors be detected, STAR would be greatly appreciate being informed of them.
- The above notwithstanding, STAR can assume no responsibility for any errors in this manual.

©Copyright 1986 Star Micronics Co., Ltd.

Table of Contents

Chapter 1	Introduction	1
	Installing the interface cartridge	
	Connecting the printer	
	Installing the ribbon and paper	
	Extra functions with the control panel	
	Self-tests	
	Hex dump	
	Panel mode	
	NLQ italic mode	
	NLQ italic and Panel mode	
	Setting print start position	
	Setting the top of form	
	Setting the left and right margins	
Chapter 2	Basic Printing	9
	Some basics of BASIC	
	A new language!	
	First steps	
	ASCII codes and the CHR\$ function	
	Control codes	
	The escape codes	
	A note on command syntax	
	Some special kinds of text	
	Near Letter Quality characters	
	Italic printing	
	Underlining	
	Superscripts and subscripts	
	Changing the print pitch	
	Expanded print	
	Making words stand out	
	Mixing print modes	

Chapter 3	Formatting Text	25
	Lines and line spacing	
	Starting a new line	
	Forward and reverse line feeds	
	Changing the line spacing	
	Moving down the page without a carriage return	
	Page control	
	Form feed	
	Reverse form feed	
	Changing the page length	
	Setting the top of form	
	Vertical form unit	
	Setting of VFU	
	Using the VFU	
	Setting left margin	
	Horizontal tabs	
	Centering and aligning text	
Chapter 4	Special Features of the Printer	41
	Now hear this	
	Resetting the printer	
	Printing the bottom of the sheet	
	Backspace and cancel text	
	Printing zeroes	
	Uni-directional printing	
	International character sets	
	Printing BIG characters	
	The optional sheet feeder	
	The macro control code	
	Reading a hex dump	
Chapter 5	Creating Your Own Characters	53
	Dot matrix printing	
	The print matrix	
	Defining your own characters	
	Rule 1: Draft download characters are eight dots high	
	Rule 2: Dots cannot overlap	
	Add up each column of dots	
	Assigning a value to your character	
	Download character definition command	
	Printing download characters	
	Defining proportional characters	
	Defining NLQ download characters	

Chapter 6	Dot Graphics	73
	Comparing dot graphics with download characters	
	Using the dot graphics commands	
	Specifying the graphics data	
	Combining text and graphics	
	Printing a design or logo	
	Plotting with your printer	
	How the program works	
	High resolution graphics	
Appendix A	DIP Switch Settings	85
	Switch functions	
Appendix B	ASCII Codes and Conversion Chart	89
Appendix C	Character Fonts	97
Appendix D	Function Codes	111
	Commands to control print style	
	Font style controls	
	Font pitch controls	
	Special print modes	
	Controlling the vertical print position	
	Line feed and related commands	
	Form feed and related commands	
	Controlling the horizontal print position	
	Download character commands	
	Dot graphics commands	
	Macro instruction commands	
	Other commands	
Appendix E	Command Summary in Numeric Order	145
Appendix F	Technical Specifications	149
Appendix G	Serial Interface Specifications	151

CHAPTER 1

INTRODUCTION

Subjects we'll cover in Chapter 1 include—

- **Installing the interface cartridge;**
- **Connecting the printer;**
- **Installing the ribbon and paper;**
- **Extra functions with the control panel.**

This interface cartridge contains all the electronics your printer needs to talk to a computer. To use your printer with a different computer, just install a different cartridge. There is an interface (I/F) cartridge for each popular computer on the market.

This interface cartridge offers full compatibility with Apple II c or any Apple computers or other serial interface computers on the market. It combines the speed and efficiency of a proven winner with the character sets and printer codes used by the Apple II c!

INSTALLING THE INTERFACE CARTRIDGE

If you have the correct interface cartridge for your computer, we can start by turning the printer around. Facing the back of it, you'll notice an opening at the left end. This is where the cartridge goes.

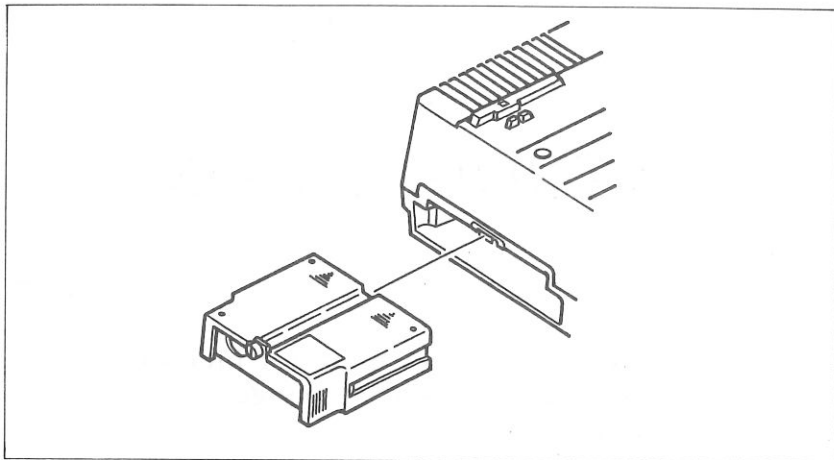


Figure 1-1. Slide the interface cartridge into the slot and fix it with a screw.

Fit the interface cartridge into the space as shown, with the round connector toward you and slide it in all the way (don't force it). If the connector is seated snugly in its socket, you should be able to tighten the screw easily. This done, connect the cable from your computer and you're ready to go.

CONNECTING THE PRINTER

Now that you have assembled your printer, it's time to use it for what you bought it for — print information from your computer. But first you have to connect it to your computer. Please follow the instructions in the order listed below.

1. Make sure both your computer and printer are turned off.
2. Connect one end of the 8-pin DIN cable to the round socket at the right rear of the printer as shown in Figure 1-2. This cable is "keyed," so it is impossible for you to connect it incorrectly. This means that you should be able to plug in the cable with only a slight pressure — don't force it! If you have to force it, it's in the wrong way. Push too hard and you'll ruin your cable.
3. Connect the other end of the cable to your computer as described in the computer manual.

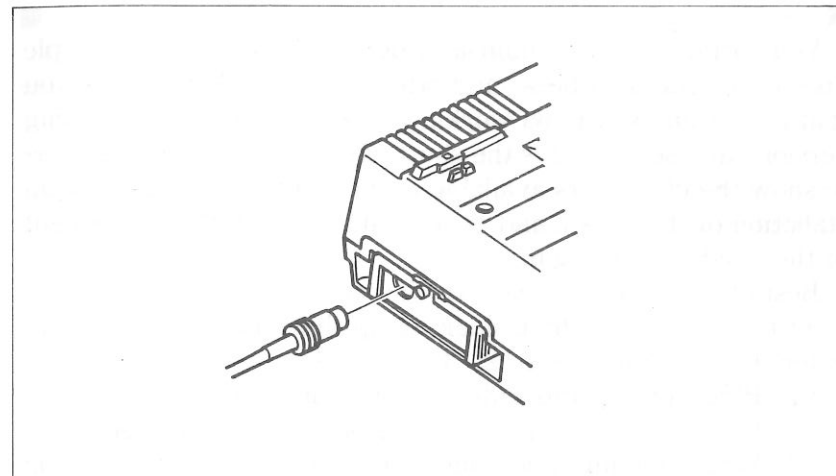


Figure 1-2. This is how you connect the cable.

INSTALLING THE RIBBON AND PAPER

After you have connected the cable, it is necessary to install the ink ribbon and the paper. They are described in your printer's manual, so please follow the procedures.

EXTRA FUNCTIONS WITH THE CONTROL PANEL

There are many functions that are not directly specified on the control panel. In this section, we'll show you these extra functions.

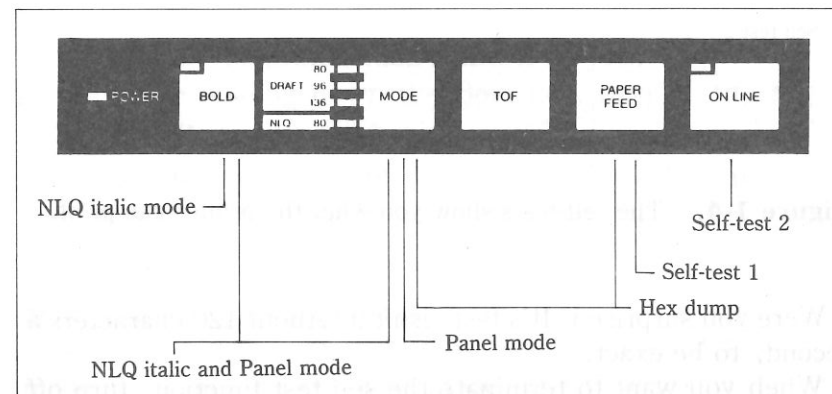


Figure 1-3. Extra functions while turning on the printer.

1. Plug in the printer (don't turn it on yet).
2. While holding down both the Mode and Bold keys, turn on the power switch.

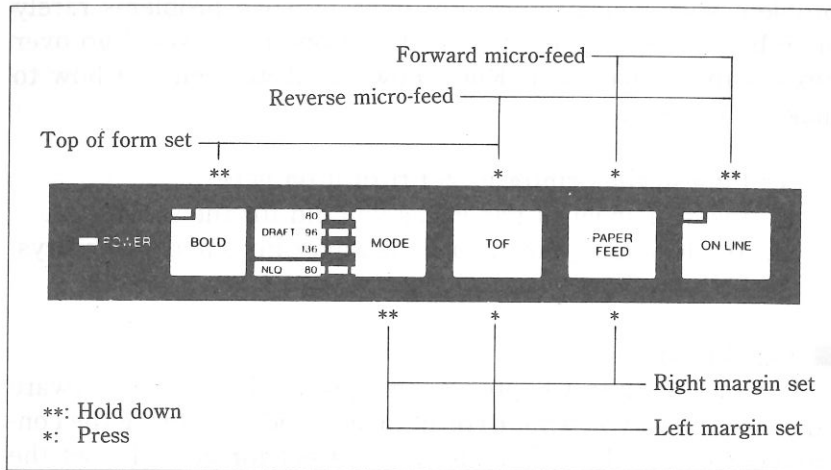


Figure 1-5. You can set many functions by the combinations of the control panel keys while in the Off Line mode.

■ Setting print start position

When you want to align the print start position, you can set it by the micro-feed operation with the control panel, instead of turning the platen knob manually.

1. Set the Off Line mode by pressing the On Line key.
2. While holding down the On Line key, press one of the following keys.

Paper Feed key — Forward micro-feed
 TOF key — Reverse micro-feed

3. When you can set the print start position, release the Paper Feed key or the TOF key first, then release the On Line key.

■ Setting the top of form

When you turn on the printer, the top of form is automatically set to the current position. If you want to change the position, you can re-set it by the following procedures.

1. Set the Off Line mode by pressing the On Line key.
2. While holding the Bold key, press the TOF key. Your printer acknowledges the new top of form with the sound of beep.

■ Setting the left and right margins

As you'll learn in Chapter 3, you can set the left margin with control code. In addition, you can set the left margin and the right margin manually by the following procedures.

1. Set the Off Line mode by pressing the On Line key.
2. While holding the Mode key, press one of the following keys.

TOF key — Left margin set
 Paper Feed key — Right margin set

3. While holding the two keys, the print head moves across the page step-by-step.
4. When the print head goes to the position where you want to set margin, release the two keys. So the printer acknowledges the margin with the sound of beep.

MEMO

CHAPTER 2

BASIC PRINTING

Subjects we'll cover in Chapter 2 include—

- Listing BASIC programs on the printer;
- How a program prints things;
- Control codes, escape codes, and command syntax;
- Near letter quality (NLQ) characters;
- Fixed and proportional character spacing;
- Special printing—
 - Printing in italics,
 - Underlining,
 - Superscripts and subscripts,
 - Boldface and emphasized text,
 - Mixing print modes.

To show you how to control your printer from a program, we choose the Applesoft BASIC because it is easy to learn and easy to use. Also, more personal computer users program in BASIC than in any other language.

The rest of this manual will show you a little BASIC — just enough for you to use your printer. We're not going to try to make you an expert programmer, though, only get you started. There are many excellent books that will teach you BASIC, so if you discover that you like to program you should have no trouble learning more about it.

SOME BASICS OF BASIC

- A new language!

Many people who meet BASIC for the first time are intimidated. Some are put off by the idea of learning any new

language, perhaps recalling the rigors of high-school Latin. Others are unnerved by anything having to do with computers.

Well don't be! In the first place, BASIC may well be the easiest language you could learn: it has a vary limited vocabulary, a simple but precise grammar, and its dialects — unlike those of English — usually different from each other only in minor detail. Without programming, a computer is a useless collection of chips and wire — why should we think of it as something special? Anyway, computers are here to stay — let's accept them with good grace.

■ First steps

The first things that a beginner learns to do are to list a program and to print a character string. Certainly these are the easiest operations one can do, but even they may depend on what computer you have. In Applesoft BASIC, we can list all the steps in a program by entering LIST. This lists them on the CRT screen; if we want to print them on a printer, we first enter the following statement to our computer.

```
PRINT CHR$(4) "PR#1"
```

This statement tells the Apple to send everything to the printer, the LIST command sends it.

When the listing to the printer has completed, enter the following statement—

```
PRINT CHR$(4) "PR#0"
```

This statement returns output to the screen.

The Applesoft BASIC command for outputting information is PRINT. Like the LIST command, this displays the information on the CRT screen so we have to add PR#1 and PR#0 if we want to use the printer.

We started with Applesoft BASIC and the programs in this manual are written in Applesoft BASIC because it is used with the Apple IIc computers. But if strange things happen when you try to run a program, check the BASIC manual that came with your computer.

Other computers, notably the Commodore C-64, require you to open the printer as a numbered device then to direct output to that device. For example, you might type the following to print "Hellow!" or to list a program in the C-64's memory.

```
OPEN4,4      OPRN4,4
CMD4         CMD4
LIST         PRINT "Hellow!"
CLOSE4      CLOSE4
```

Here, the first line says that the printer is device number 4, the second directs output to it, the third does the printing, and the last line closes down device number 4.

Now that we know how to address the printer, let's try listing a BASIC program. Load a program into memory ready to program printer operation — just as soon as we learn a little bit about the ASCII codes.

■ ASCII codes and the CHR\$ function

You talk to your computer in BASIC, but your computer and your printer talk to each other in what are known as ASCII codes. In the ASCII code, each number from 0 to 255 has a particular meaning — 36, for example, makes the printer print a dollar sign. Some numbers cause the printer to do other things, too. For instance, sending a 7 sounds the printer's bell.

Taken together, these numbers and their meanings make up the ASCII code (pronounced *ask-key*), which stands for the *American Standard Code for Information Interchange*. There are ASCII codes for all the letters of the alphabet (upper case and lower case), 0 to 9, most punctuation marks, and some (but not all) of the functions of the printer.

There are a number of different ways to represent an ASCII code, depending on how you are using it. For example, the ASCII codes for the letter "A" are 65 (decimal) or &H41 (hexadecimal). Or you can just call it "A". Appendix B shows all of the ASCII codes.

BASIC uses the CHR\$ function to represent ASCII characters and many functions. To print the letter "A" we would enter PRINT CHR\$(65). To make the printer's bell sound, we would PRINT CHR\$(7). In general, we print a character by entering PRINT CHR\$(*ASCII code*) to the printer after "PR#1" statement.

We can also use hex ASCII codes. Although we use only decimal ASCII codes in this manual, you should understand at least what a hex code is. "Hex" is short for hexadecimal and refers to a base-16 number (the numbers we use in everyday life are base 10). Since the hex system needs 16 digits, it uses the numerals 0 through 9 and also the letters A through F. You can always tell that a number is in hexadecimal by the "&H" immediately preceding it. The ASCII code for the letter "A" (65 in decimal) is &H41 in hex.

■ Control codes

ASCII codes with values of 32 or less do not have their own keys. These codes control many of the printer's functions, so we call them control codes. To enter a control code from the keyboard, we have to press two keys at the same time — the "control" key and one other. The other key determines what code is sent — pressing the control (CTRL) key and A sends ASCII code 1, CTRL B sends ASCII code 2, and so on.

Your printer has a lot of control codes to let you do some really nifty things. Let's try one that we've mentioned several times already:

```
10 REM Demo of ASCII code
20 LPRINT CHR$(4) "PR#1"
30 LPRINT CHR$(7)
40 LPRINT CHR$(4) "PR#0"
RUN
```

That's the printer's bell (we call it that even though it sounds like a buzzer). We'll learn more about it in later (we just wanted to show you a control code that would get your attention right away).

There are four common ways of referring to a control code: the name of the code or its abbreviation, the decimal ASCII value, the hexadecimal ASCII value, and the "CTRL—" value. For example, the ASCII code that causes the printer to advance the paper one line is decimal 10. This code may be referred to by any of the following.

line feed	— the name of the code
<LF>	— its abbreviation

ASCII 10	— its decimal value
ASCII &H0A	— its hexadecimal value (the &H signifies hex)
CHR\$(10)	— the way it's used in BASIC
CTRL-J	— the way you send it from a keyboard

Of course, most of the time we don't need to bother with these. Our computers are smart enough to know that when we press the "A" key we want to print the letter "A" — they take care of all the intermediate steps.

Appendix B is a table that shows the various names for each code so you can convert back and forth. The microcomputer world is not very consistent in describing ASCII codes, so it's important that you have a basic knowledge of them.

■ The escape codes

Back when the ASCII system was set up, computer equipment was relatively simple and thirty-three control codes were considered sufficient at the time. The American Standards people realized that, eventually, more control codes would be needed so they included the escape (ESC) code to allow almost any number of additional codes to be defined when they became necessary.

ESC allows us to "escape" from the ordinary set of control codes so we can specify additional functions and other information needed for a printer function. In this manual, we'll write the ESC code inside broken brackets, like this — <ESC>.

<ESC> — decimal 27 — is always followed by at least one other number; it is never used alone. The whole series of related numbers is called an escape sequence.

■ A note on command syntax

Because the readers of this manual will be running such a wide variety of applications on so many different computers, we just can't show the exact way of sending codes to the printer for each one of them. Instead, as we introduce you to each new command, we will show the commands as in this example:

<ESC> "a" 2

This command that turns on the NLQ printing. <ESC>, as we mentioned earlier, is the escape code (which is ASCII code 27).

A letter or number in quotes (such as the "a" above) means that the character should be sent to the printer (without the quotes). In our example, you should send a letter following the escape code. In BASIC, you could do this in a couple of ways: by sending the character itself (e.g. PRINT "a"); or by using the CHR\$ function to send the ASCII code for the character (e.g. PRINT CHR\$(97);).

When shown as in the above example (i. e. no quotes and no "CHR\$"), you can use either ASCII code 2 (i. e. CHR\$(2)) or the character "2" (which is ASCII code 50). The same idea applies to commands ending with 1 or 0.

So for our example above, any of these BASIC statements will have the same result:

```
PRINT CHR$(27);"a";CHR$(2)
PRINT CHR$(27);"a";CHR$(50)
PRINT CHR$(27);"a2"
```

Even though, there are many commands that require the use of ASCII code 0; the character "0" (ASCII code 48) cannot be substituted. In these cases, instead of an unadorned 0 we will show CHR\$(0) each time these commands are referenced.

That's it for the basics. You are now ready to learn how to use the many features of your printer.

SOME SPECIAL KINDS OF TEXT

If you looked carefully at your printer's self-test, you noticed that it can print in italics. But there's more! Your printer can underline characters, print superscripts and subscripts, and perhaps most exciting, print near letter quality characters.

■ Near Letter Quality characters

This printer's Near Letter Quality (sometimes abbreviated as NLQ) character set is ideal for correspondence and other important printing, for it takes a keen eye to detect that it is from a dot matrix printer. Normally, your printer prints draft quality characters. This is adequate for most work and it prints fastest. But for the final printout, try NLQ. The program below shows how.

```
10 REM Demo of NLQ character set
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(27);"a2"
40 PRINT "This line shows NEAR LETTER QUALITY!"
50 PRINT CHR$(27);"a1";
60 PRINT "This line shows standard print."
70 PRINT CHR$(4) "PR#0"
```

In this program, line 30 selects NLQ characters with <ESC> "a" 2 command. Line 40 prints a sample before line 50 switches printer back to draft printing with an <ESC> "a" 1. When you run the program you should get this:

```
This line shows NEAR LETTER QUALITY!
This line shows standard print.
```

Table 2-1
Near letter quality commands

Function	Control code
Near letter quality ON	<ESC> "a" 2
Near letter quality OFF	<ESC> "a" 0 or <ESC> "a" 1

■ Italic printing

Italic letters are letter that are slanted to the right. Your printer can print all of its letters in *Italic* as well as the roman (standard) letters you are accustomed. Italics can be used to give extra emphasis to certain words. The control codes to turn italic on and off are shown in Table 2-2.

Table 2-2
Italic commands

Function	Control code
Italic ON	<ESC> "4"
Italic OFF	<ESC> "5"

Use this program to see italic characters:

```
10 REM Demo of italic and roman
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(27);"4";
```

```

40 PRINT "This line is in ITALIC characters."
50 PRINT CHR$(27);"a2";
60 PRINT "This line is in NLQ ITALIC characters."
70 PRINT CHR$(27);"5";
80 PRINT "This line is in NLQ ROMAN characters."
90 PRINT CHR$(27);"a1";
100 PRINT "This line is in Draft ROMAN characters."
110 PRINT CHR$(4) "PR#0"

```

Here is what you should get:

```

This line is in ITALIC characters.
This line is in NLQ ITALIC characters.
This line is in NLQ ROMAN characters.
This line is in Draft ROMAN characters.

```

In this program, line 30 turns italic on with `<ESC> "4"`, and line 50 selects the NLQ characters. Line 70 turns italic off with `<ESC> "5"`, and line 90 cancels the NLQ characters.

■ Underlining

Not only can your printer print all styles of printing in both roman and italic, but it can underline them too. The control codes are shown in Table 2-3.

Table 2-3
Underline commands

Function	Control code
Underline ON	<code><ESC> "X"</code>
Underline OFF	<code><ESC> "Y"</code>

Again, that's simple. Let's try it with this program:

```

10 REM Demo of underlining
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(27);"X";
40 PRINT "This phrase is UNDERLINED;";
50 PRINT CHR$(27);"Y";
60 PRINT " this is not."
70 PRINT CHR$(4) "PR#0"

```

It should come out like this:

```

This phrase is UNDERLINED; this is not.

```

In this program underline is turned on in line 30 with `<ESC> "X"`, and then off in line 50 with `<ESC> "Y"`. There's a new little wrinkle in this program, though. The semicolons at the end of the first three lines told BASIC that those lines were to be continued. Therefore, BASIC didn't send a carriage return and line feed at the end of those lines. We just did this to illustrate that all these control codes can be used in the middle of a line. It's easy to underline or *italicize* only part of a line.

■ Superscripts and subscripts

Your printer can print in two different heights of characters. The smaller characters are called *superscripts* and *subscripts* and are half the height of normal characters. *Superscripts* print even with the tops of regular printing while *subscripts* print even with the bottom of regular printing. They are frequently used to reference footnotes, and in mathematical formulas.

Table 2-4 has the codes for using superscripts and subscripts.

Table 2-4
Superscripts and subscripts commands

Function	Control code
Superscript ON	<code><ESC> "x" or <ESC> "w"</code>
Subscript ON	<code><ESC> "y"</code>
Super and subscript OFF	<code><ESC> "z" or <ESC> "W"</code>

Try this program to see them work:

```

10 REM Demo of superscripts and subscripts
20 PRINT CHR$(4) "PR#1"
30 PRINT "Look! ";
40 PRINT CHR$(27);"x";
50 PRINT "SUPERSCRIPTS ";
60 PRINT CHR$(27);"z";
70 PRINT "& ";
80 PRINT CHR$(27);"y";
90 PRINT "SUBSCRIPTS ";

```

```
100 PRINT CHR$(27);"z";
110 PRINT "on one line."
120 PRINT CHR$(4) "PR#0"
```

Look! `SUPERSCRIPTS & SUBSCRIPTS ON ONE LINE.`

Here line 40 turns on superscripts with `<ESC> "x"`. It's turned off in line 60 with `<ESC> "z"`. Then between printing text, subscripts are turned on in line 80 with `<ESC> "y"`, and finally off in line 100. Again, everything prints on one line because of the semicolons.

CHANGING THE PRINT PITCH

In "printer talk," the number of characters that can be printed in one inch is called the *print pitch* or *character pitch*. Normally, your printer is set for 10 characters per inch, which is called *pica* (and is the same as the pica pitch on some typewriters). This works out to 80 characters per line.

You can also print 12 characters per inch (*elite pitch*). This gives you 96 characters per line.

Condensed print is approximately 17 characters per inch. Condensed pitch allows you to get 136 columns of printing on an 8-1/2 inch page.

Proportional spacing provides an alternative to the block-style output of a defined pitch. It moves its print head only as far as each character needs. Thus, the print head moves further for "M"s and "W"s than for "I"s and "i"s.

You can set these pitches by using the Mode key on the control panel manually, or by the software as shown in the table below.

Table 2-5
Print pitch commands

Pitch	Characters/inch	Control code
Pica	10	<code><ESC>"N" or <ESC>"n"</code>
Elite	12	<code><ESC>"E"</code>
Condensed	17	<code><ESC>"e" or <ESC>"Q" or <ESC>"q"</code>
Pica proportional	—	<code><ESC>"p"</code>
Elite proportional	—	<code><ESC>"P"</code>

Try this program to see how the print pitches work:

```
10 REM Demo of print pitches
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(27);"Q";
40 PRINT "This line shows CONDENSED pitch."
50 PRINT CHR$(27);"E";
60 PRINT "This line shows ELITE pitch."
70 PRINT CHR$(27);"P";
80 PRINT "This line shows ELITE PROPORTIONAL
pitch."
90 PRINT CHR$(27);"p";
100 PRINT "This line shows PICA PROPORTIONAL
pitch."
110 PRINT CHR$(27);"N";
120 PRINT "This line shows PICA pitch."
130 PRINT CHR$(4) "PR#0"
```

When you run this program, you should get this:

```
This line shows CONDENSED pitch.
This line shows ELITE pitch.
This line shows ELITE PROPORTIONAL pitch.
This line shows PICA PROPORTIONAL pitch.
This line shows PICA pitch.
```

Line 30 turns on condensed print with `<ESC> "Q"`. Line 40 prints a line at 17 characters per inch. Line 50 turns on elite pitch with `<ESC> "E"`. Line 60 prints a line at 12 characters per inch. The `<ESC> "P"` in line 70 selects the elite proportional and line 80 prints a line with elite proportional spacing. The `<ESC> "p"` in line 90 selects the pica proportional and line 100 prints a line with pica proportional spacing. Finally, the `<ESC> "N"` in line 110 resets the printer to pica pitch and line 120 prints a line in pica pitch.

NOTE: When you change the print pitch by the MODE key on the control panel, these proportional pitches should be automatically cancelled.

■ Expanded print

Each of the print pitches can be enlarged to twice its normal

width. This is called expanded print. Try this program to see how it works:

```
10 REM Demo of expanded print
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(14);
40 PRINT "Expanded mode stays"
50 PRINT "on until it is ";
60 PRINT CHR$(15);
70 PRINT "turned off."
80 PRINT CHR$(4) "PR#0"
```

```
Expanded mode stays
on until it is turned off.
```

Expanded print set with CHR\$(14) stays on until you turn it off with CHR\$(15). Note that you didn't need to put an <ESC> in front of the CHR\$(14) or CHR\$(15).

Table 2-6
Expanded print commands

Function	Control code
Expanded ON	CHR\$(14)
Expanded OFF	CHR\$(15)

MAKING WORDS STAND OUT

Your printer has very good print density when it's just printing regularly. But sometimes you may want something to stand out from the rest of the page. This printer provides two ways to do this: boldface and emphasized print. Both of these go over the characters twice, but they use slightly different methods to darken the characters. Let's try them and see what the difference is.

The following table shows the control codes for getting into and out of boldface and emphasized modes.

Table 2-7
Print emphasis commands

Function	Control code
Boldface ON	<ESC> "[
Boldface OFF	<ESC> "]"
Emphasized ON	<ESC> "!"
Emphasized OFF	<ESC> CHR\$(34)

Try them now with this little program:

```
10 REM Demo of boldface and emphasized
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(27); "[";
40 PRINT "This line is BOLDFACE printing."
50 PRINT CHR$(27); "!";
60 PRINT "This line is BOLDFACE and EMPHASIZED."
70 PRINT CHR$(27); "]"";
80 PRINT "This line is EMPHASIZED printing."
90 PRINT CHR$(27); CHR$(34);
100 PRINT "This line is normal printing."
110 PRINT CHR$(4) "PR#0"
```

Run this program. The results will look like this:

```
This line is BOLDFACE printing.
This line is BOLDFACE and EMPHASIZED.
This line is EMPHASIZED printing.
This line is normal printing.
```

Line 30 turns on boldface with <ESC> "[", and line 40 prints a line of text. In line 50 emphasized is turned on with <ESC> "!". Line 60 prints a line of text in boldface *and* emphasized. Line 70 then turns boldface off with <ESC> "]" so that line 80 can print in emphasized only. Finally, line 90 turns emphasized off, so your printer is set for normal printing.

Look closely at the different lines of printing. In the line of boldface printing each character has been printed twice, and they are moved down just slightly the second time they are printed. In emphasized printing, they are moved slightly to the right the second time your printer prints. The last line combined both of these so that each character was printed 4 times. Now that's pretty nice printing, isn't it?

MIXING PRINT MODES

We have learned how to use the various print modes individually and together. Now we'll see how to combine them more efficiently.

You have at your disposal a unique command that lets you choose any valid combination of print modes and pitch. This is the Master Print mode command. It looks like this:

`<ESC> "i" n`

Here, the value of n defines the print style to be selected. The value of n can range from 0 to 255, which is the range of values that can be stored in one eight-bit byte. If you look at each bit in this byte, you'll find that each one represents a printing style variation. Adding the binary values of the selected bits gives the value of n for a particular combination of print styles.

Table 2-8 shows the decimal values of the bits in the Master Print byte. To calculate the value n for a particular combination of printing styles, just add the values of the features that you want to combine.

Table 2-8
Values of mixing print styles for Master Print

Bit	Print style	Decimal value
1	Elite print	1
2	Proportional print	2
3	Condensed print	4
4	Emphasized print	8
5	Boldface print	16
6	Expanded print	32
7	(Not used)	
8	Underline	128

For example, if you want to select elite expanded boldface print, you would calculate the value of n like this:

Elite	1
Boldface	16
Expanded	32
$n =$	<u>49</u>

The command would look like this:

`CHR$(27);"i";CHR$(49)`

To better understand the way the print modes work, consider that each mode except pica (pica is the default) has a separate switch that can be turned on and off via software. Once the switch is on, it stays on until turned off. When two modes that conflict are turned on at the same time, the printer must choose which one to use.

For example, suppose you turn on both Elite and Condensed modes. Since these cannot combine the printer must make a choice; in this case, the printer chooses Elite.

Summary notes

- 1) Pica is the default pitch and is active when Elite or Condensed is turned off.
- 2) When two modes conflict, the one of lesser priority is cancelled. For example, Condensed and Emphasized cannot be printed at the same time, printing is Emphasized.
- 3) Elite cancels Emphasized and Condensed.
- 4) Underline, and Expanded modes combine with any print modes.
- 5) Emphasized will not mix with Elite or Condensed.

MEMO

CHAPTER 3

FORMATTING TEXT

Subjects we'll cover in Chapter 3 include—

- The carriage return and line feed;
- The amount of space between lines;
- Moving to the next page;
- The number of lines on a printed page;
- Horizontal and vertical tabs;
- Setting margins — left, right, top and bottom;
- Centering and aligning.

Chapter 2 showed us all the basic techniques of using the printer. Now we're ready for the more advanced ones. We'll concentrate on changing the appearance of the page to suit our needs.

LINES AND LINE SPACING

■ Starting a new line

Up until now the only time we have thought about printing on a new line is when we *didn't* want it to happen. We learned that putting a semicolon (;) at the end of a BASIC line will *not* end the line of printing. So somehow, the computer telling the printer when to end one line and start another.

There are two codes that are used to end one line and start another. They are *carriage return* (CHR\$(13)) and *line feed* (CHR\$(10)). Like the escape code, they have been given abbreviations which you'll find many texts (including this one): <CR> and <LF>. The codes are simple, but their action is a little confusing (especially with BASIC). Carriage return is the easiest. Each time that the printer receives a CHR\$(13) it returns the print head to the left margin. It does not advance the paper.

Line feed is more complicated. Each time the printer receives a CHR\$(10) it both advances the paper one line and returns the print head to the left margin, ready to start a new line (if special command is not sent to the printer; see below).

■ Forward and reverse line feeds

Your printer has a unique capability: it can move the paper up or down! Its unique tractor design allows the paper to be fed in either direction without jamming. This allows you to move around the page at will. You can use this feature to print several columns of text side by side, or print a graph and then move back up and insert descriptive legends. As you experiment you're bound to come up with more uses!

The simplest form of reverse paper feeding is a reverse line feed. The code is <ESC> "r", which sets the paper to move down (in effect, moving the printing *up*) one line. A "line" used in a reverse line feed is the same size as a line in a regular line feed (this is normally 1/6 inch). When you change the line spacing (which you'll read about next), you change it for both forward and reverse line feeds. The code is <ESC> "f", when you want to reset the reverse line feed.

Table 3-1
Line feed commands

Function	Control code
Return print head to left margin	CHR\$(13)
Advance paper one line	CHR\$(10)
Select reverse line feed	<ESC> "r"
Cancel reverse line feed	<ESC> "f"

■ Changing the line spacing

When you turn your printer on the line spacing is set to 6 lines per inch. This is fine for most printing applications, but sometimes you may want something different. Your printer makes it easy to set the line spacing to whatever you want.

Try this program to see how easy it is to change the line spacing:

```
10 REM Demo of line spacing
20 PRINT CHR$(4) "PR#1"
```

```
30 FOR I=1 TO 25
40 IF I<10 THEN A$="0"+RIGHT$(STR$(I),1):GOTO 60
50 A$=RIGHT$(STR$(I),2)
60 PRINT CHR$(27);"T";A$;
70 PRINT "THIS LINE SPACING IS SET TO ";A$;"/144
   INCH."
80 NEXT I
90 PRINT CHR$(27);"A";
100 PRINT "LINE SPACING IS SET TO 1/6 INCH
   (NORMAL)."
```

```
110 PRINT CHR$(4) "PR#0"
```

This is what you will get:

```
THIS LINE SPACING IS SET TO 1/144 INCH.
THIS LINE SPACING IS SET TO 2/144 INCH.
THIS LINE SPACING IS SET TO 3/144 INCH.
THIS LINE SPACING IS SET TO 4/144 INCH.
THIS LINE SPACING IS SET TO 5/144 INCH.
THIS LINE SPACING IS SET TO 6/144 INCH.
THIS LINE SPACING IS SET TO 7/144 INCH.
THIS LINE SPACING IS SET TO 8/144 INCH.
THIS LINE SPACING IS SET TO 9/144 INCH.
THIS LINE SPACING IS SET TO 10/144 INCH.
THIS LINE SPACING IS SET TO 11/144 INCH.
THIS LINE SPACING IS SET TO 12/144 INCH.
THIS LINE SPACING IS SET TO 13/144 INCH.
THIS LINE SPACING IS SET TO 14/144 INCH.
THIS LINE SPACING IS SET TO 15/144 INCH.
THIS LINE SPACING IS SET TO 16/144 INCH.
THIS LINE SPACING IS SET TO 17/144 INCH.
THIS LINE SPACING IS SET TO 18/144 INCH.
THIS LINE SPACING IS SET TO 19/144 INCH.
THIS LINE SPACING IS SET TO 20/144 INCH.
THIS LINE SPACING IS SET TO 21/144 INCH.
THIS LINE SPACING IS SET TO 22/144 INCH.
THIS LINE SPACING IS SET TO 23/144 INCH.
THIS LINE SPACING IS SET TO 24/144 INCH.
THIS LINE SPACING IS SET TO 25/144 INCH.
LINE SPACING IS SET TO 1/6 INCH (NORMAL).
```

Line 60 changes the line spacing. The <ESC> "T" d_2 command changes the line spacing to $d_2/144$ of an inch. The loop that is started in line 30 increases the value of d_2 (the variable A\$ in this program) each time it is executed. So the line spacing increases as the program continues. Finally, the <ESC> "A" in line 90 resets the line spacing to 6 lines per inch.

You may wonder why they picked 1/144 of an inch as the increment for the line spacing command. There's a good reason: the dots that the printer makes are 1/72 inch apart. So this means that you can vary the line spacing in increments as fine as one half dot spacing.

Table 3-2 shows all the line spacing commands, including several "shortcut" commands for commonly used line spacings.

Table 3-2
Line spacing commands

Function	Control code
Set line spacing to 1/6 inch	<ESC> "A"
Set line spacing to 1/8 inch	<ESC> "B"
Set line spacing to $d_2/144$ inch	<ESC> "T" d_2
Set line feed without carriage return	<ESC> "I" 1
Set normal line feed function	<ESC> "I" 0

■ Moving down the page without a carriage return

So far, all the commands that move the paper also move the print head to the left margin. And normally this is what you want. Sometimes, though, you may wish to move down the page without moving the print head back to the left margin. The following commands do just that.

The <ESC> "I" 1 command selects the printer to move down the page without moving the print head to the left margin. The <ESC> "I" 0 command restores the normal line feed function. Try this program to see how they work:

```
10 REM Demo of line feed fur
20 PRINT CHR$(4) "PR#1"
30 PRINT "LINE NUMBER 1."
40 PRINT "LINE NUMBER 2.";
50 REM Line feed without carriage return
60 PRINT CHR$(27);"11";CHR$(10);
70 PRINT "LINE NUMBER 3."
80 PRINT "LINE NUMBER 4."
90 PRINT CHR$(4) "PR#0"
```

Here is what your printer will produce:

```
LINE NUMBER 1.
LINE NUMBER 2.
                LINE NUMBER 3.
LINE NUMBER 4.
```

The <ESC> "I" 1 command in line 60 sets the printer to move the page without carriage return. So line 70 prints line just

below the previous line. Notice that both line 40 and 60 end with semicolons.

PAGE CONTROL

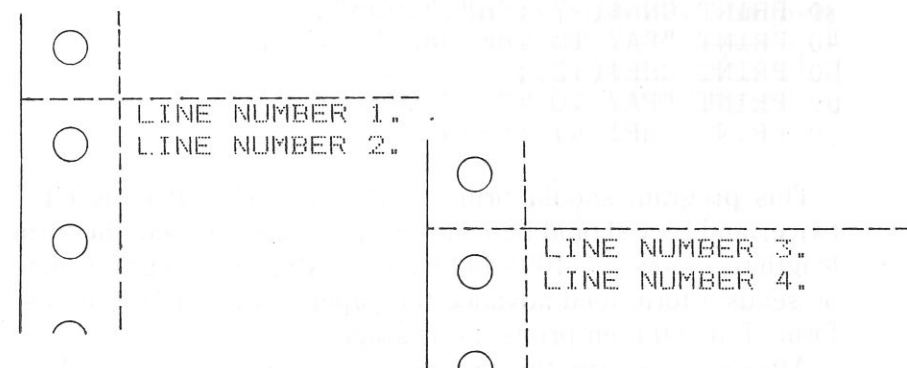
Now that we have seen how to control line spacing, we can go on to page control — positioning the printing on the page and adjusting the paper length.

■ Form feed

The simplest forms control code is the *form feed*. Form feed (or <FF>) is CHR\$(12) and causes the printer to move the paper to the top of the next sheet. Try it by changing lines 50 and 60 to this:

```
50 REM Form feed
60 PRINT CHR$(12);
```

Before you run the program, turn your printer off and adjust the paper so that the top of the sheet is even with the top of the ribbon guide on the print head, then turn the printer back on. If you don't remember how to do this, review the separated manual. When you run the program, the results will look like this:



The form feed (CHR\$(12)) in line 60 caused the printer to move to the top of a new page before printing the last two lines.

■ Reverse form feed

Just as your printer can perform a reverse line feed, it can do a reverse form feed. This code moves the paper so that the print

head is positioned at the top of the current page. This can be used, for example, to print text in a multi-column magazine format; print the first column, then reverse form feed back to the top of the page to start the second column. To make the reverse form feed, enter the reverse line feed command ($\langle \text{ESC} \rangle$ "r") and then enter $\langle \text{FF} \rangle$.

Table 3-3
Form feed commands

Function	Control code
Advance paper to top of next page	CHR\$(12)
Reverse paper to top of current page	$\langle \text{ESC} \rangle$ "r" CHR\$(12)
Set the page length to $d_4/144$ inches	$\langle \text{ESC} \rangle$ "H" d_4
Set the top of form to the current position	$\langle \text{ESC} \rangle$ "v"

■ Changing the page length

You may have some computer forms that you wish to use with this printer that are not 11 inches high. That's no problem, because you can tell your printer how high the forms are that you are using. Let's set up a 7 inches high form length, which is typical of many computer checks. The following program will do it.

```
10 REM Demo of variable form length
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(27);"H";"1008";
40 PRINT "PAY TO THE ORDER OF:";
50 PRINT CHR$(12);
60 PRINT "PAY TO THE ORDER OF:"
70 PRINT CHR$(4) "PR#0"
```

This program should print "PAY TO THE ORDER OF:" twice, and they should be 7 inches apart. Line 30 sets the form length to 7 inches (=1008/144 inches). After line 40 prints, line 50 sends a form feed advance the paper to the top of the next form. Line 60 then prints its message.

After you have run this program, turn off the printer and adjust the top of form position. When you turn the printer back on the page length will reset to its normal setting (usually 11 inches).

■ Setting the top of form

When you turn on the printer, the top of form is automatically

set to the current position. If you want to change the position, you can re-set it by the following control code:

$\langle \text{ESC} \rangle$ "v"

In addition, you can re-set it with the control panel as described in Chapter 1.

VERTICAL FORM UNIT

This interface cartridge contains a control code called a Vertical Form Unit (or VFU). This command allows setting of the page length, vertical tab positions, and the bottom margin with one control code.

■ Setting of VFU

To set VFU, send $\langle \text{GS} \rangle$ code to start setting, then "A@" code (see below) to represent TOF (top of form), "n@" code to represent vertical tab position, "A@" code to represent TOF on the next page, and $\langle \text{RS} \rangle$ code to end setting of VFU. VFU has channel A to channel F. The channel A sets TOF position and channels B to F set each independent vertical tab. The relation between the setting position of these channels and data code is shown below:

Table 3-4
Relationship between the data bits and the VFU channels

Bit	8	7	6	5	4	3	2	1
2-byte data	0	1	CH.F	CH.E	CH.D	CH.C	CH.B	CH.A
	0	1	0	0	0	0	0	0

In this table, if the tab is to be set for CH.A to CH.F, 1 is specified, if not, 0. When the bottom margin is to be set, setting for both CH.A and CH.B is 1.

Let's practice setting! Setting of VFU at power-on is described with the above control codes.

At power on:

1. The form length is set to 11 inches (66 lines).
2. The vertical tab is set every 6 lines after the 7th line for channel B and not set for other channels.
3. The bottom margin is not set.

These conditions are shown below.

Channel No.	F	E	D	C	B	A	Remarks		
Data Bit	8	7	6	5	4	3	2	1	
Start Definition	0	0	0	1	1	1	0	1	GS code
1st line	0	1	0	0	0	0	0	0	TOF position
2nd line	0	1	0	0	0	0	0	0	
3rd line	0	1	0	0	0	0	0	0	
4th line	0	1	0	0	0	0	0	0	
5th line	0	1	0	0	0	0	0	0	
6th line	0	1	0	0	0	0	0	0	
7th line	0	1	0	0	0	0	1	0	Tab set
8th line	0	1	0	0	0	0	0	0	
9th line	0	1	0	0	0	0	0	0	
10th line	0	1	0	0	0	0	0	0	
11th line	0	1	0	0	0	0	0	0	
12th line	0	1	0	0	0	0	0	0	
13th line	0	1	0	0	0	0	1	0	Tab set
14th line	0	1	0	0	0	0	0	0	
15th line	0	1	0	0	0	0	0	0	
16th line	0	1	0	0	0	0	0	0	
17th line	0	1	0	0	0	0	0	0	
18th line	0	1	0	0	0	0	0	0	
19th line	0	1	0	0	0	0	1	0	Tab set

<GS> code is "00011101" in binary, which is 29 in decimal and &H1D in hexadecimal according to the table in Appendix B.

Similarly, the TOF position is "01000001" and "01000000" in binary, 65 and 64 in decimal and &H41 and &H40 in hexadecimal. They correspond to "A" and "@" in ASCII code.

To replace these tables with a program in this way:

```

10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(29);
30 PRINT "A@";"@";"@";"@";"@";"@";"@";"@";
40 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
50 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
60 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
    
```

```

70 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
80 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
90 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
100 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
110 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
120 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
130 PRINT "B@";"@";"@";"@";"@";"@";"@";"@";
140 PRINT "A@";
150 PRINT CHR$(30)
160 PRINT CHR$(4) "PR#0"
    
```

A program to set a bottom margin to the 15th line is written in which the page length is 3 inches (18 lines). The vertical tab is set to 6th, 12th, and 17th lines for channel B, 4th and 10th lines for channel C, 7th and 14th lines for channel D, 2nd, 5th, 8th, and 11th lines for channel E.

These settings are summarized in the table below:

Channel No.	F	E	D	C	B	A	Remarks		
Data Bit	8	7	6	5	4	3	2	1	
Start Definition	0	0	0	1	1	1	0	1	GS code
1st line	0	1	0	0	0	0	0	1	TOF position
2nd line	0	1	0	0	0	0	0	0	Tab set for channel E
3rd line	0	1	0	0	0	0	0	0	
4th line	0	1	0	0	0	1	0	0	Tab set for channel C
5th line	0	1	0	1	0	0	0	0	Tab set for channel E
6th line	0	1	0	0	0	0	1	0	Tab set for channel B
7th line	0	1	0	0	1	0	0	0	Tab set for channel D
8th line	0	1	0	1	0	0	0	0	Tab set for channel E
9th line	0	1	0	0	0	0	0	0	

10th line	0	1	0	0	0	1	0	0	Tab set for channel C
11th line	0	1	0	1	0	0	0	0	Tab set for channel E
12th line	0	1	0	0	0	0	1	0	Tab set for channel B
13th line	0	1	0	0	0	0	0	0	
14th line	0	1	0	0	1	0	0	0	Tab set for channel D
15th line	0	1	0	0	0	0	1	1	Bottom margin set
16th line	0	1	0	0	0	0	0	0	
17th line	0	1	0	0	0	0	1	0	Tab set for channel B
18th line	0	1	0	0	0	0	0	0	
1st line (Next page)	0	1	0	0	0	0	0	1	TOF position
End Definition	0	0	0	1	1	1	1	0	RS code

To write programs same as the above.

```

10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(29);
30 PRINT "A@";"P@";"@";"D@";"P@";"B@";
40 PRINT "H@";"P@";"@";"D@";"P@";"B@";
50 PRINT "@@";"H@";"C@";"@";"B@";"@";
60 PRINT "A@";
70 PRINT CHR$(30)
140 PRINT CHR$(4) "PR#0"
    
```

Table 3-5
VFU commands

Function	Control code
Start VFU definition	<GS>
Set TOF position	"A@"
Set BOF position	"C@"
Set no tab	"@@"
Set tab in channel B	"B@"
Set tab in channel C	"D@"
Set tab in channel D	"H@"
Set tab in channel E	"P@"
Set tab in channel F	"~@"
Set tab in channels B and C	"F@"
Set tab in channels B and D	"J@"
Set tab in channels B and E	"R@"
Set tab in channels B and F	"b@"
Set tab in channels C and D	"L@"
Set tab in channels C and E	"T@"
Set tab in channels C and F	"d@"
Set tab in channels D and E	"X@"
Set tab in channels D and F	"h@"
Set tab in channels E and F	"p@"
Set tab in channels B, C and D	"N@"
Set tab in channels B, C and E	"V@"
Set tab in channels B, C and F	"f@"
Set tab in channels B, D and E	"Z@"
Set tab in channels B, D and F	"j@"
Set tab in channels B, E and F	"r@"
Set tab in channels C, D and E	"\@"
Set tab in channels C, D and F	"l@"
Set tab in channels C, E and F	"t@"
Set tab in channels D, E and F	"x@"
Set tab in channels B, C, D and E	"^@"
Set tab in channels B, C, D and F	"n@"
Set tab in channels B, C, E and F	"v@"
Set tab in channels B, D, E and F	"z@"
Set tab in channels C, D, E and F	" @"
Set tab in channels B, C, D, E and F	"~@"
End VFU definition	<RS>

■ Using the VFU

To feed to the tab position, send <US> *d* code to the printer. For example, to run tab setting for channel B, add the following programs.

```
80 FOR I=1 TO 5
90 PRINT "* Channel B *";
100 PRINT CHR$(13);
110 PRINT CHR$(31);"B";
120 NEXT I
130 PRINT
```

When you run this program, you should get like this:

```
* Channel B *
```

```
* Channel B *
```

```
* Channel B *
```

```
* Channel B *
```

```
* Channel B *
```

When you check the printout sample, you'll find that no printing at the tab position on the 17th line, though the tab is set to 6th, 12th, 15th, and 17th for channel B. Because the bottom

margin is set to the 15th line and the printer ignores the tab position on the 17th line and moves to next page. In this way, no printing is made in an area from the bottom margin to the next TOF.

Table 3-6
VFU using commands

Function	Control code
Feed to next tab in channel B	<US> "B"
Feed to next tab in channel C	<US> "C"
Feed to next tab in channel D	<US> "D"
Feed to next tab in channel E	<US> "E"
Feed to next tab in channel F	<US> "F"
Feed to next BOF or TOF	<US> "A"

SETTING LEFT MARGIN

The left margin of this printer works just like a typewriter — once it is set all the printing is done between the left margin and the right end. The command to set the left margin is given in the following table:

Table 3-7
Left margin command

Function	Control code
Set left margin at column d_3	<ESC> "L" d_3

Try setting the margin with this program:

```
10 REM Demo of left margin
20 PRINT CHR$(4) "PR#1"
30 GOSUB 80
40 PRINT CHR$(27);"L";"020";
50 GOSUB 80
60 PRINT CHR$(4) "PR#0"
70 END
80 FOR I=1 TO 80
90 PRINT "X";
100 NEXT I
110 PRINT
120 RETURN
```

The first thing that this program does is to branch to the subroutine that starts in line 80. This subroutine prints 80 X's in a row. The first time that the subroutine is used, all the X's fit in one line. Then line 40 sets the left margin to 20. Once again the subroutine is used, but this time the X's won't all fit on one line since there is now only room for 60 characters between the margins.

Run the program. The results will look like this:

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

When you want to reset the left margin to the default value, you have two choices. You can either turn the printer off and back on, or you can set margin value equal to the default value. This means that you should set a left margin of 0.

If you change the pitch of your printing after you set your margin, the margin will not change. It stays at the same place on the page. So if you set the margin to give you 65 columns of printing when you are using pica type, then you change to elite type you will have room for more than 65 columns of elite printing between the margins.

HORIZONTAL TABS

Suppose you need to move across the page to a certain position several times in a document. It's not much fun to type in space after space. And you don't have to — you can "tab" your way across the page.

Your printer's tabs are like those on a typewriter, but much more powerful. You have horizontal tabs which can be used for both text and graphics — and they're really handy for indenting paragraphs and making tables.

Horizontal tabs are not set as the power-on default.

Try this program to see how the horizontal tabs work.

```
10 REM Demo of horizontal tabbing
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(9);CHR$(1)
```

```
40 PRINT CHR$(27);"(";"007,014,021,028,035."
50 PRINT "ONE";CHR$(9);"TWO";CHR$(9);"THREE";
  CHR$(9);"FOUR"
80 PRINT CHR$(4) "PR#0"
```

Here's what you should get—

```
ONE      TWO      THREE    FOUR
```

<ESC> "(" is the command to begin setting horizontal tabs. It must be followed by characters representing the positions that you want the tabs set. In our program we are setting tabs in columns 7, 14, 21, 28 and 35. The comma (,) indicates to separate each tab position, and the period (.) at the end ends the string of tabs. In fact, any character that is not greater than the previous one will stop setting tabs. This means that you must put all your tab values in order, from least to greatest, or they won't all get set.

Now, add the following lines to the program:

```
60 PRINT CHR$(27);")";"014,021."
70 PRINT "ONE";CHR$(9);"TWO";CHR$(9);"THREE";
  CHR$(9);"FOUR"
```

When you run the program now it produces this:

```
ONE      TWO      THREE    FOUR
```

```
ONE      TWO                                THREE    FOUR
```

<ESC> ")" is the command to cancel horizontal tabs. It must be followed by characters representing the positions that you want the tabs clear. In our program we are clearing the tabs in columns 14 and 21. The comma (,) indicates to separate each tab position, and the period (.) at the end ends the string of tabs. In fact, any character that is not greater than the previous one will stop clearing tabs. This means that you must put all your tab values in order, from least to greatest, or they won't all get clear.

When you want to clear the all tabs, turn your printer off and on again or enter the <ESC> "0" code to the printer.

Table 3-8
Horizontal tab commands

Function	Control code
Advance to next tab position	CHR\$(9)
Set tabs at $d1_3$, $d2_3$, etc.	<ESC> "(" $d1_3$ "," $d2_3$ ","" ," dx_3 "."
Add a tab at $d1_3$	<ESC> "u" $d1_3$
Clear tabs at $d1_3$, $d2_3$, etc.	<ESC> ")" $d1_3$ "," $d2_3$ ","" ," dx_3 "."
Clear all tabs	<ESC> "0"

CENTERING AND ALIGNING TEXT

Text can be arranged in any of three formats: left aligned (normal printing with "ragged" right end), centered between the margins, or right aligned. These are selected by the following commands.

Table 3-9
Aligning commands

Function	Control code
Left-aligned printing	<ESC> "J" CHR\$(0)
Centered printing	<ESC> "J" CHR\$(1)
Right-aligned printing	<ESC> "J" CHR\$(2)

Try this program to see how easy it is.

```
10 REM Demo of aligning and centering
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(27);"L";"040";
40 PRINT CHR$(27);"J";CHR$(0);
50 PRINT "THIS LINE IS LEFT-ALIGNED."
60 PRINT CHR$(27);"J";CHR$(1);
70 PRINT "THIS LINE IS CENTERED."
80 PRINT CHR$(27);"J";CHR$(2);
90 PRINT "THIS LINE IS RIGHT-ALIGNED."
100 PRINT CHR$(4) "PR#0"
```


When you run this program, you should get like this:

```
THIS LINE IS LEFT-ALIGNED.  
    THIS LINE IS CENTRED.  
                THIS LINE IS RIGHT-ALIGNED.
```

CHAPTER 4

SPECIAL FEATURES OF THE PRINTER

Subjects we'll cover in Chapter 4 include—

- **Printer's bell;**
- **Master reset;**
- **Uni-directional printing;**
- **International character sets;**
- **Printing BIG characters;**
- **The optional sheet feeder;**
- **Macro instruction;**
- **Reading a hex dump.**

In the previous chapters we have learned about several groups of control codes. In this chapter we will look at more control codes. These codes don't fit neatly into any of the groupings that we have studied, but they add a lot of capability to your printer. So here goes.

■ Now hear this

You may have heard the printer's *bell* if you have ever run out of paper. And you may have wondered why it's called a bell when it *beeps* instead of ringing! It's a long story that goes back to the early days of computers, when teletype machines were used for computer terminals. These mechanical marvels had a bell in them that could be heard for blocks. This bell was used to signal the operator that somethings needed attention. The code that the computer sent to the teletype machine to ring the bell was, reasonably enough, called a *bell code*. Well the name *bell code* is still with us, even if the bell has changed to a beeper, and a lot of people still call the beeper a bell, even if it doesn't sound like one. So with our trivia lesson out of the way, let's see how we can "ring the bell."

The code to sound the "bell" is `CHR$(7)`, which is ASCII code 7 or `<BEL>`. Any time your printer receives this code it will

sound the bell for a quarter of a second. This can be used to remind an operator to change the paper or to make another adjustment to the printer.

You can try this by typing:

```
PRINT CHR$(7)
```

■ Resetting the printer

Up to now when we wanted to reset the printer to the power on condition we have had to either turn the printer off and then on again, or to send the specific codes that reset the particular features. There is an easier way. The control code `<ESC> "c"` will reset all of the printer's features to the power on condition (as determined by the DIP switches), with two exceptions. Those exceptions are that `<ESC> "c"` will not erase any characters that you have stored in the printer's RAM memory (Chapter 5 tells you how to create your own characters), and it won't erase the macro if you have one stored in the printer's RAM (this chapter will tell you how to create a macro).

In addition, if you set the "Panel" mode, "NLQ italic" mode, or "NLQ Italic and Panel" mode by the control panel settings at the power on, these functions will remain with this control code.

■ Printing the bottom of the sheet

Sometimes when you are using individual sheets of paper you may want to print near the bottom of a sheet. The paper-out detector usually stops the printer when you are about 1 inch from the bottom of the sheet. This is to notify you if you are running out of continuous paper.

Your printer has the ability to print right to the bottom of the sheet. You can disable the paper-out detector so that it doesn't stop the printer. This will allow you to print to the end of the sheet, and even beyond if you are not careful. The codes to control the paper-out detector, along with the other codes that we have just learned are in the following table.

Table 4-1
Some miscellaneous commands

Function	Control code
Sound bell	CHR\$(7)
Master reset	<ESC> "c"
Paper-out detector off	<ESC> "O"
Paper-out detector on	<ESC> "o"
Move print head back one space	CHR\$(8)
Cancel text in print buffer	CHR\$(24)
Print "slash zero"	<ESC> "D" CHR\$(0) CHR\$(1)
Print "normal zero"	<ESC> "Z" CHR\$(0) CHR\$(1)

■ Backspace and cancel text

Backspace (CHR\$(8)) "backs up" the print head so that you can print two characters right on top of each other. Each time your printer receives a backspace it moves the print head one character to the left, instead of to the right. You can *strike over* multiple letters by sending more than one backspace code.

Cancel text (CHR\$(24)) deletes all the text in the print buffer; that is, in the line before the delete text command. Since your printer prints one line of text at a time, only that line will be deleted.

The following program shows how these codes work.

```
10 PRINT CHR$(4) "PR#1"
20 PRINT "BACKSPACE DOES NOT";
30 PRINT CHR$(8);CHR$(8);CHR$(8);
40 PRINT "=== WORK."
50 PRINT "CANCEL LINE";
60 PRINT CHR$(24);
70 PRINT "DOES NOT WORK."
80 PRINT CHR$(4) "PR#0"
```

Here is what this program will print:

```
BACKSPACE DOES NOT WORK.
DOES NOT WORK.
```

The backspace codes in line 30 move the print head a total of three spaces to the left so that the first part of line 40 will overprint the word "NOT".

In line 60, CHR\$(24) deletes the words in line 50. The semicolon at the end of line 50 prevents a line feed from causing that line to print before the printer receives the CHR\$(24) code. The text in line 70 prints as it normally would because it is after CHR\$(24).

■ Printing zeroes

Believe it or not, there are two types of zeroes. There is of course the type we use every day — 0 — and this is what your printer will print if you don't do anything.

The other type is used almost exclusively in computers and engineering. It is called the "slash zero" and is written like this — Ø. The line through the number is supposed to prevent you from misreading it as the letter "O". Back before high-quality printers were available, this was a good idea but you really have no need for it (although you may want to use the slash zero for special effect).

■ Uni-directional printing

Uni-directional printing is big word that means *printing in one direction only*. Your printer normally prints when the print head is moving in both directions. But once in a while you may have an application where you are more concerned about how the vertical lines align than with how fast it prints. This printer lets you make this choice. The table below shows the commands for controlling how this printer prints.

Table 4-2
Printing direction commands

Function	Control code
Print in one direction	<ESC>">"
Print in both directions	<ESC>"<"

Try this program to see the difference that printing in one direction makes.

```
10 REM Demo of uni-directional printing
20 PRINT CHR$(4) "PR#1"
30 PRINT CHR$(27);"T";"14";
40 FOR I=1 TO 10
50 PRINT "| "
60 NEXT I
```

```
70 PRINT : PRINT
80 PRINT CHR$(27);">";
90 FOR I=1 TO 10
100 PRINT "| "
110 NEXT I
120 PRINT CHR$(12);CHR$(27);"c"
130 PRINT CHR$(4) "PR#0"
```

Here is what you will get. The top line is printed bi-directionally, and the bottom is printed uni-directionally. You will have to look hard because there isn't much difference.

Let's analyze the program. Line 30 sets the line spacing to 14/144 of an inch so that the characters that we print will touch top to bottom. Lines 40 ~ 60 print 10 vertical line characters. Then line 80 sets one-direction printing and the vertical lines are printed again. Finally line 120 sends a form feed to advance the paper to the top of a new page, and then uses the master reset to restore the printer to the power on condition.

■ International character set

Your printer is a multi-lingual printer for it can speak in eight languages! Your printer changes languages by changing 10 characters that are different for the different languages. These sets of characters are called *international character sets*. The control codes to select the international character sets are given in Table 4-3.

Table 4-3
International character set commands

Country	Control code
U.S.A.	<ESC>"Z"CHR\$(7) CHR\$(0)
Italy	<ESC>"Z"CHR\$(6) CHR\$(0) <ESC>"D"CHR\$(1) CHR\$(0)
Denmark	<ESC>"Z"CHR\$(5) CHR\$(0) <ESC>"D"CHR\$(2) CHR\$(0)
England	<ESC>"Z"CHR\$(4) CHR\$(0) <ESC>"D"CHR\$(3) CHR\$(0)
Germany	<ESC>"Z"CHR\$(3) CHR\$(0) <ESC>"D"CHR\$(4) CHR\$(0)
Sweden	<ESC>"Z"CHR\$(2) CHR\$(0) <ESC>"D"CHR\$(5) CHR\$(0)
France	<ESC>"Z"CHR\$(1) CHR\$(0) <ESC>"D"CHR\$(6) CHR\$(0)
Spain	<ESC>"D"CHR\$(7) CHR\$(0)

The characters that change are shown beneath their ASCII code in Table 4-4.

Table 4-4
International character sets

Country	35	64	91	92	93	96	123	124	125	126
U. S. A.	#	@	[\]	~	{		}	~
Italy	£	§	°	ç	é	ù	à	ò	è	ì
Denmark	#	@	Æ	Ø	À	~	æ	ø	å	~
England	£	@	[\]	~	{		}	~
Germany	#	§	Ä	Ö	Ü	~	ä	ö	ü	ß
Sweden	#	@	Ä	Ö	À	~	ä	ö	å	~
France	£	à	°	ç	§	~	é	ù	è	..
Spain	£	§	ì	Ñ	¿	~	•	ñ	ç	~

■ Printing BIG characters

You can even enlarge your character sets for attention-grabbing headings or special effects. There are two commands you can use. Everything following any of them will be enlarged as shown below, until the cancel code is entered.

Table 4-5
Big character commands

Function	Control code
Double-high enlarged print	<ESC> "h" CHR\$(1)
Quad-high enlarged print	<ESC> "h" CHR\$(2)
Cancel enlarged print	<ESC> "h" CHR\$(0)

Try this program to see the big characters.

```

10 REM Demo of big characters
20 PRINT CHR$(4) "PR#1"
30 PRINT "THIS IS ";
40 PRINT CHR$(27);"h";CHR$(1);
50 PRINT "DOUBLE";
60 PRINT CHR$(27);"h";CHR$(0);
70 PRINT " SIZED PRINTING."
80 PRINT "THIS IS ";
90 PRINT CHR$(27);"h";CHR$(2);
100 PRINT "QUAD";
110 PRINT CHR$(27);"h";CHR$(0);
120 PRINT " SIZED PRINTING."
130 PRINT CHR$(4) "PR#0"

```

When you run this program, you will get like this:

```

THIS IS DOUBLE SIZED PRINTING.
THIS IS QUAD SIZED PRINTING.

```

As you can see, when the big character command is used, the baseline for each character does not align. When you want to align the baseline, try this program:

```

10 REM Demo of aligning big characters
20 PRINT CHR$(4) "PR#1"
30 PRINT "THIS IS ";
40 PRINT CHR$(27);"T16";CHR$(27);"11";
50 PRINT CHR$(27);"r";CHR$(10);
60 PRINT CHR$(27);"h";CHR$(1);
70 PRINT "DOUBLE";
80 PRINT CHR$(27);"h";CHR$(0);
90 PRINT CHR$(27);"T02";CHR$(10);
100 PRINT " SIZED PRINTING.";CHR$(27);"f";
    CHR$(27);"A"
110 PRINT : PRINT : PRINT
120 PRINT "THIS IS ";
130 PRINT CHR$(27);"T48";CHR$(27);"r";CHR$(10);
140 PRINT CHR$(27);"h";CHR$(2);

```

```

150 PRINT "QUAD";
160 PRINT CHR$(27);"h";CHR$(0);
170 PRINT CHR$(27);"T08";CHR$(10);
180 PRINT " SIZED PRINTING.";CHR$(27);"f";
    CHR$(27);"A"
190 PRINT CHR$(27);"c"
200 PRINT CHR$(4) "PR#0"

```

When you run this program, you will get like this:

THIS IS **DOUBLE** SIZED PRINTING.

THIS IS **QUAD** SIZED PRINTING.

■ The optional sheet feeder

The automatic sheet feeder is a handy option that feeds single cut sheets automatically. Work done on cut sheets looks better that done on computer paper, and you don't have to tear the "ears" off each sheet as you must with fan-fold paper.

The automatic sheet feeder feeds a new sheet automatically every time the printer receives or generates a form feed. Any time you wish, you can turn the auto-feed unit on and off by using control codes.

Table 4-6
Automatic sheet feeder commands

Function	Control code
Select automatic feed mode	<ESC> CHR\$(25) CHR\$(4) or "((4))"
Cancel automatic feed mode	<ESC> CHR\$(25) CHR\$(0) or "((0))"
Insert paper	<ESC> CHR\$(25) CHR\$(1) or "((1))"
Eject paper	<ESC> CHR\$(25) "R" or "((R))"

When the automatic sheet feeder is installed, you must set the DIP switch 1-2 on to detect the paper-out condition.

In addition, following functions are ignored when the automatic sheet feeder is installed:

- Setting of the page length
- Vertical format units
- Reverse feed settings

■ The macro control code

The last of our group of miscellaneous codes is definitely not the least. It is a *user-defined* control code, called a *macro* control code. The term *macro* is from the jargonese *macro-instruction* which refers to an instruction that "calls," or uses a group of normal instructions. In computer programming macro-instructions (which are similar to subroutines) save programmers a lot of time and effort. Your printer's macro can save you a lot of time and effort also.

Here is how the printer's macro works. You *define macro* by telling the printer what normal control codes are to be included in the macro. Then you can use the macro any time that you want and the printer will do all the things that you included the macro definition. You can include up to 16 codes in a single macro. You can even use the macro to store a frequently used word or phrase. There are two control codes for the macro: one to define it, and one to use it. They are given in the Table 4-7.

Table 4-7
Macro instruction commands

Function	Control code
Define macro	<ESC>"+" ... (codes you include) ... CHR\$(30)
Use macro	<ESC>"+"CHR\$(1)

To see how this works we can build a macro that will reset the printing style to normal, no matter what style it may be to start with. The following program will define a macro to do this.

```

10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(27);"+"; :REM Start macro
30 PRINT CHR$(27);"h";CHR$(0); :REM Big character
                                off

```

```

40 PRINT CHR$(27);"f";           :REM Normal feed
50 PRINT CHR$(27);"i";CHR$(0);   :REM Select normal
                                pica
60 PRINT CHR$(27);"z";           :REM Super &
                                subscripts off
70 PRINT CHR$(27);"A";           :REM Set 1/6 inch
                                line spacing
80 PRINT CHR$(27);"J";CHR$(0);   :REM Left-aligned
                                printing
90 PRINT CHR$(30)                 :REM End macro
100 PRINT CHR$(4) "PR#0"

```

As the comments in the program, we started to define macro in line 20. Line 30 cancels the big character printing. Line 40 sets the normal line feed direction. Line 50 sets the normal pica, and also this command cancels the proportional pitch, condensed print, expanded print, boldface, emphasized, and the underlining. Line 60 cancels the superscripts and the subscripts. Line 70 sets the line spacing to 1/6 inch, and line 80 sets the left-aligned printing. Then, line 90 ends the macro definition. This printer will remember this macro until the power is turned off or until a new macro is defined. A macro can hold up to 16 bytes (characters) of information. The one that we defined contains fifteen.

Now that you have defined a macro, let's see how to use it. This program will print one line using several printing features. Then it "calls" the macro in line 70. When line 80 prints the style is "plain vanilla" because the macro has reset it.

```

10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(27);"L040";
30 PRINT CHR$(27);"J";CHR$(2);
40 PRINT CHR$(27);"X";
50 PRINT CHR$(27);"h";CHR$(1);
60 PRINT "TESTING ABCD"
70 PRINT CHR$(27);"+";CHR$(1);
80 PRINT "TESTING ABCD"
90 PRINT CHR$(4) "PR#0"

```

TESTING ABCD

TESTING ABCD

■ Reading a hex dump

We've seen how to make a hex dump in Chapter 1, but it's not really clear what we can do with one. We need a little background first.

The BASIC in some computers changes ASCII codes before they send them to the printer. If you run into problem because of this, try this hex dump to check the ASCII codes.

First turn off the printer. Hold down both the Paper Feed key and Mode key and turn on the printer. Then, run the following program.

```

10 PRINT CHR$(4) "PR#1"
20 FOR I=0 TO 255
30 IF PEEK(49601)>127 THEN 30
40 POKE 49296,I
50 NEXT I
60 PRINT
70 PRINT CHR$(4) "PR#0"

```

If your system passes the codes directly to the printer without changing them, you will get like this.

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	! " # \$ % & ' () * + , - . /
30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	0123456789:;<=>?
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	@ABCDEFGHIJKLMNO
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	PQRSTUVWXYZ[\]^_
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	`abcdefghijklmnopqrstuvwxyz
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F	~`0123456789:;<=>?
80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
AA	A1	A2	A3	A4	A5	A6	A7	AB	A9	AA	AB	AC	AD	AE	AF
BA	B1	B2	B3	B4	B5	B6	B7	BB	B9	BA	BB	BC	BD	BE	BF
CA	C1	C2	C3	C4	C5	C6	C7	CB	C9	CA	CB	CC	CD	CE	CF
DA	D1	D2	D3	D4	D5	D6	D7	DB	D9	DA	DB	DC	DD	DE	DF
EA	E1	E2	E3	E4	E5	E6	E7	EB	E9	EA	EB	EC	ED	EE	EF
FA	F1	F2	F3	F4	F5	F6	F7	FB	F9	FA	FB	FC	FD	FE	FF
8D	8A															..

Your printer prints hex numbers 16 per line, with printing the characters on the right side. If it receives less than 16, it sits in a holding pattern, awaiting more data. Taking the printer off line with the On Line key dumps the characters to the paper.

To debug a program quickly, just use the hex dump capability. Appendix B will help you translate the hex codes to ASCII equivalents.

MEMO

CHAPTER 5

CREATING YOUR OWN CHARACTERS

Subjects we'll cover in Chapter 5 include—

- Designing and printing your own characters;
- Designing proportional characters;
- Designing your own characters with NLQ.

In the previous chapters of this manual you've learned how to control the printer to give dozens of different typefaces. By using various combinations of pitches, character weights, and font selections, you can create nearly any effect you want to in text. And with international character sets and the special text and big characters described in Chapter 4, you can print almost any character you think of.

But if “almost any character” isn't good enough for you, then it's a good idea you have this printer! With it you can actually create your own characters. As you'll see in this chapter, *download characters* can be used to print a logo, special characters for foreign languages, scientific and professional applications, or any other specific printing task.

DOT MATRIX PRINTING

In order to create download characters, you'll need some understanding of how dot matrix printers work. They're called “dot matrix” because each character is made up of a group of dots. Look closely at some printed characters produced by your printer and you will see the dots. Figure 5-1 shows how the letter “C” is formed by printing 15 dots.

The print head in this printer consists of nine wires stacked one atop the other. Figure 5-2 shows an enlarged schematic view of the front of the print head, showing the ends of the wires and their relationship to the printed draft characters. As you can

see, the capital letters use the top seven wires of print head, and the descenders (such as the lower case “p” shown) use the bottom seven pins. As the print head moves across the page (in either direction — that’s what is meant by bi-directional printing) it prints one column of dots at a time. Each time a dot is supposed to print an electromagnet inside the print head causes the appropriate wire to strike the ribbon (making this printer an *impact* printer).

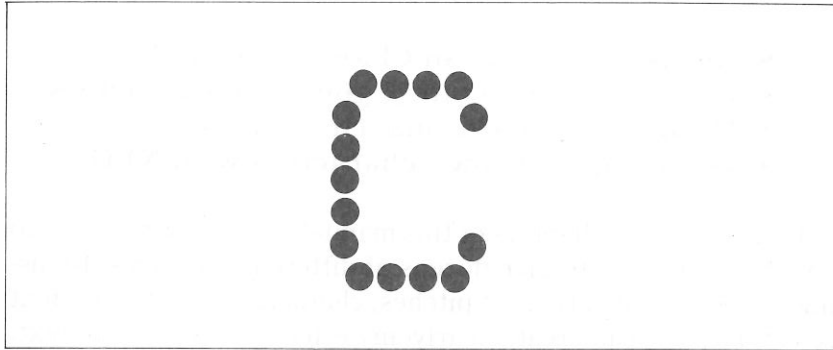


Figure 5-1. The letter “C” is created by printing 15 dots.

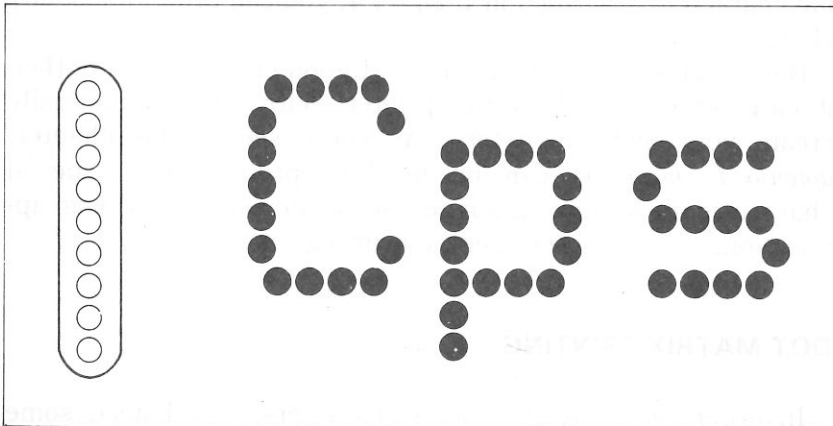


Figure 5-2. As the print head moves across the page, each of the wires prints one row of dots.

THE PRINT MATRIX

All of the standard characters that this printer prints are formed from patterns of dots that are permanently stored in the printer’s *ROM* (read-only memory). This includes all of the

standard ASCII characters, and special characters, the international character sets, and the NLQ characters.

But there is another area of memory in this printer reserved for *user-defined* characters. These are characters that you can design and *download* into the printer. When download characters are defined they are stored in *RAM* (random access memory), which allows you to define or modify them at any time.

Each of these characters, whether it is from the standard character ROM or in download RAM, is constructed on a grid which is six “boxes” by nine “boxes” high. In addition, a dot can straddle any of the vertical lines. As an example, take a look at the enlarged “9” superimposed on the grid in Figure 5-3. As you can see, some dots are inside the boxes, and some are centered on the vertical lines. This, in effect, makes the character grid 11 dots wide by 9 dots high. To see how the rest of the characters in the standard character ROM are constructed, take a look at Appendix C.

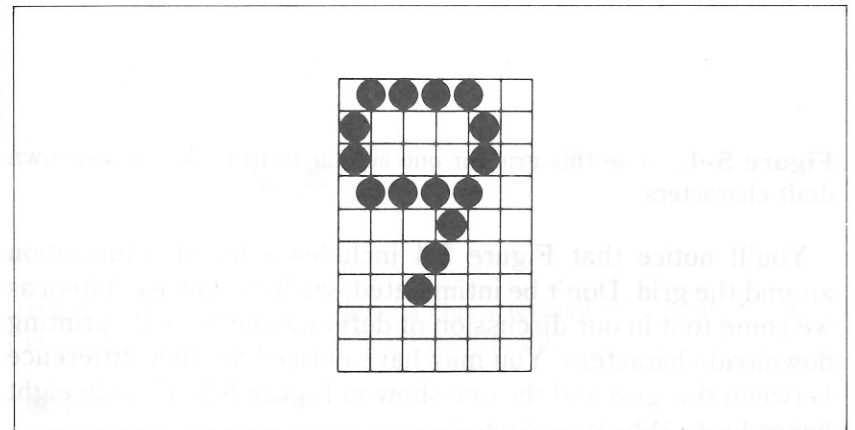


Figure 5-3. Dots can be inside boxes or straddle the vertical lines of the grid.

DEFINING YOUR OWN CHARACTERS

You’ve seen how these characters are designed by using a grid to layout the dots. Now you can define characters exactly the same way. Make up some grids (photocopy Figure 5-4 if you wish) and get ready to be creative! (Just in case you are not feeling creative, and to make our explanations a little clearer, we’ll

be using a picture of a chemist's flask as an example of a draft download character. Your can see how we've laid it out in Figure 5-5. Later in this chapter we'll use this character to create a small graph.)

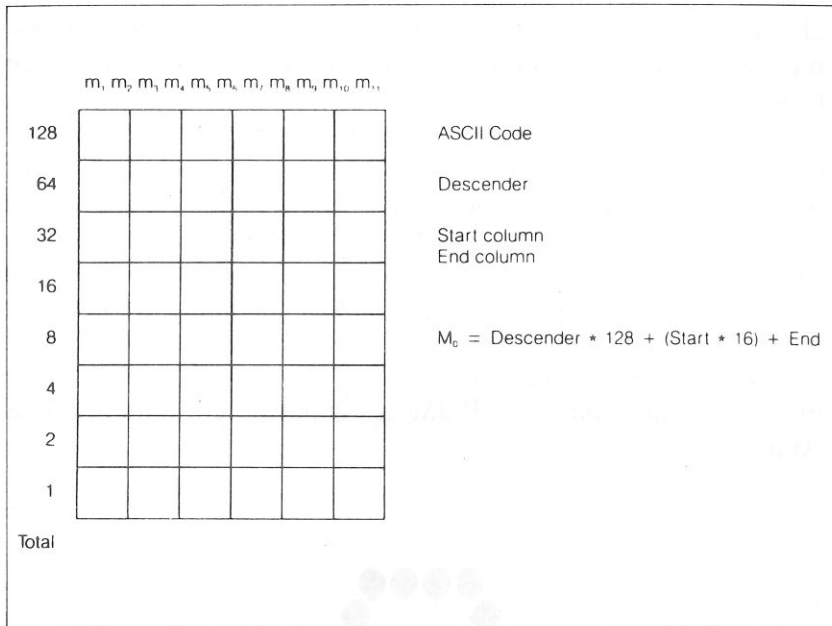


Figure 5-4. Use this grid (or one similar to it) to define your own draft characters.

You'll notice that Figure 5-4 includes a lot of information around the grid. Don't be intimidated; we'll explain each item as we come to it in our discussion of defining and actually printing download characters. You may have noticed another difference between this grid and the one show in Figure 5-3: it's only eight boxes high. Which leads us to ...

■ Rule 1: Draft download characters are eight dots high

As you noticed in Figure 5-2, capital letters, most lowercase letters, and most special characters use only the top seven pins of the print head. Draft download characters can go one better: they can use as many as eight of the nine wires in the print head. So our grid is eight dots high.

It's also possible to use the bottom eight pins, just as the "g", "j", "p", "q", and "y" of the standard character sets do. These are called descenders (because the bottom of the character

descends below the baseline of the rest of the characters).

One bit in the download character definition command is to be treated as a descender or not. We'll get to the command in due time. For now, if your character uses the top eight dots, write in a one next to the word "Descender" on the layout grid; if it uses the bottom eight dots, write in a zero. In our example, we'll want the bottom of the flask to line up with the baseline of the other characters, so it will *not* be a descender. As shown in Figure 5-5, we've written in a "1" on our grid.

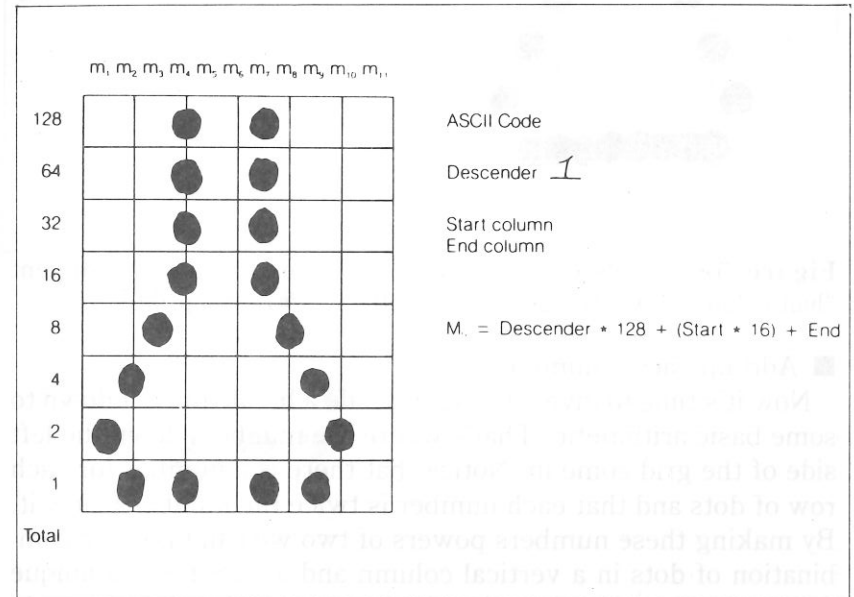


Figure 5-5. We've designed a character and decided that it would not be a descender, hence the "1" written in.

■ Rule 2: Dots cannot overlap

As you can see in Figure 5-5 our flask has a nearly continuous outline. But, you may ask, why not make it a *really* solid line and print all the intermediate dots, as shown in Figure 5-6? Because the dots that straddle the vertical lines in the grid actually overlap those inside the boxes. If we tried to print overlapping dots, the print head would have to slow down and back up to print both dots — not very efficient! To avoid this inefficiency, this printer will not allow you to define a character like Figure 5-6. (Actually, you can define it, but when it prints, your printer will leave out the overlapping dots, so that it would print like Figure 5-5.)

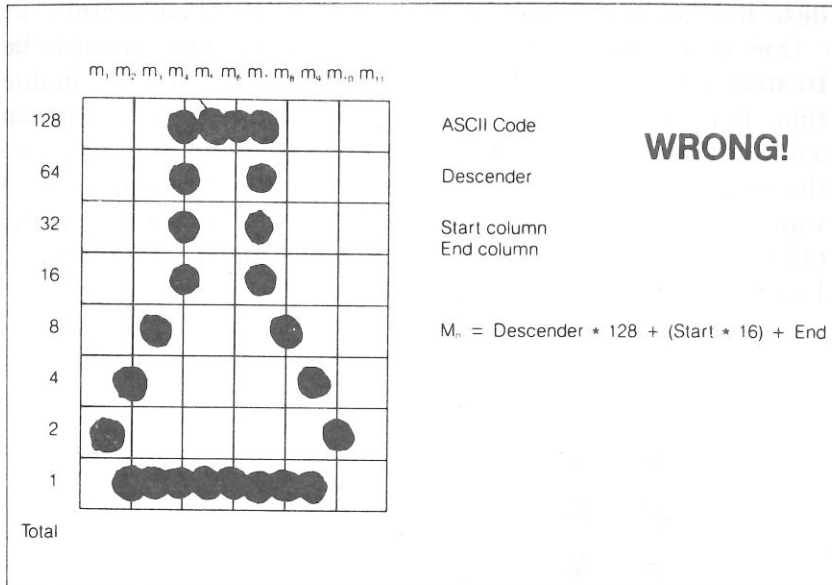


Figure 5-6. Dots cannot overlap; those in immediately adjacent “half columns” will be ignored when the character is printed.

■ Add up each column of dots

Now it’s time to give our creative side a break and get down to some basic arithmetic. That’s where the numbers down the left side of the grid come in. Notice that there is a number for each row of dots and that each number is twice the number below it. By making these numbers powers of two we can take any combination of dots in a vertical column and assign them a unique value. Some examples will make this clearer. As shown in Figure 5-7, if we add the numbers for the dots that print in a column, the sum will be a number in the range of 0 to 255. Each number from 0 ~ 255 represents a unique combination of dots.

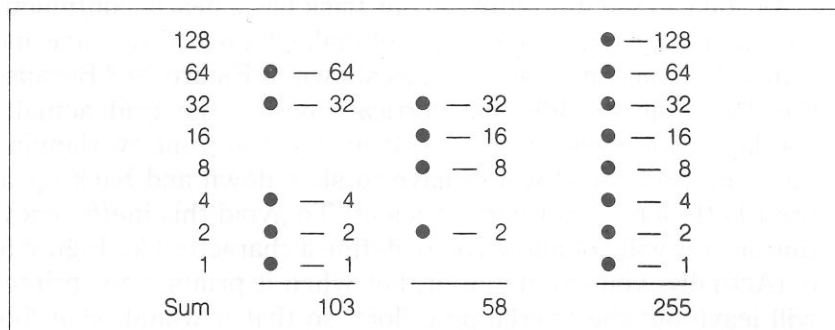


Figure 5-7. By adding the values of each dot in a column, you’ll get a unique description for any combination of dots.

So add up the values of the dots in each column using this system. In Figure 5-8 we’ve shown our grid with the sums of the columns filled in across the bottom (see if these agree with your answers!). Across the top of the grid you’ve probably noticed the cryptic labeling of each column: *m1, m2, m3*, etc. These labels correspond to the labels in the command syntax statement, which we’ll get to shortly.

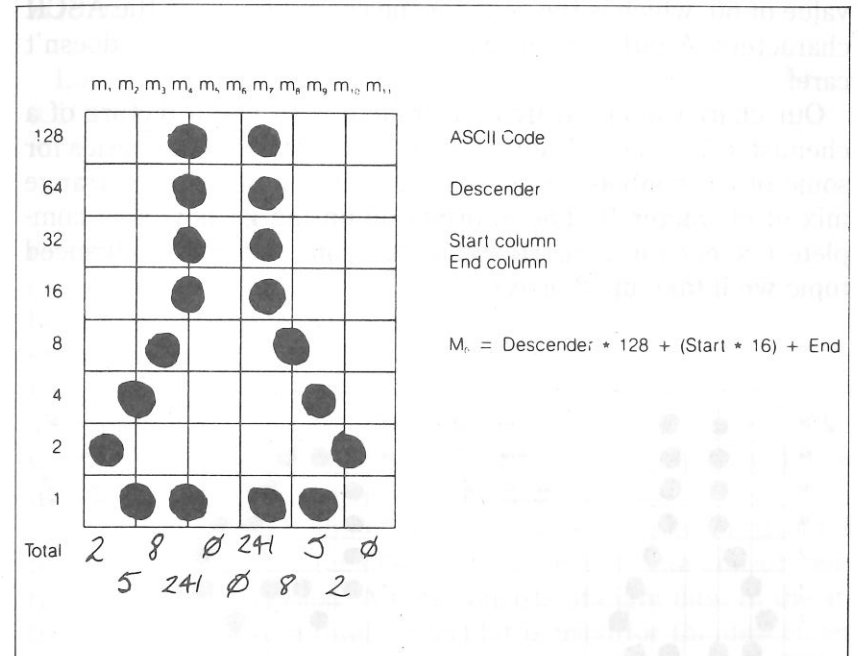


Figure 5-8. Add the values of the dots in each column and write the sum of each column at the bottom.

■ Assigning a value to your character

We’ve done a pretty thorough job of designing and describing a user-defined character. But this printer has room for 96 download characters — how does it know which standard character we want to print: every character is assigned a unique number.

The standard characters are assigned the ASCII codes — numbers from 0 to 255. For the download character sets you can define any positions between 32 to 127. This means that once a character is defined and assigned a value (and the download character set is selected). You can use that character on the printer the same way you would any standard character. You

can send the character with the same ASCII value. You can also access the character from a BASIC program with the CHR\$ function.

Except for the limitation that download characters must be between 32 and 127, there are no rules or restrictions on the use of numbers. This means you can use whatever is most convenient for you — perhaps seldom-used keys can be replaced by more useful characters. In our example, we'll assign the flask a value of 60, which is the code for the character "<" in the ASCII characters. A rather arbitrary selection, but this printer doesn't care!

Our chart would hardly be complete with just a picture of a chemist's flask, so in Figure 5-9 we've made completed grids for some other symbols: an automobile and a gun (quite a strange mix of characters!). The information on the grids is now complete (except for proportional width data — a more advanced topic we'll take up shortly).

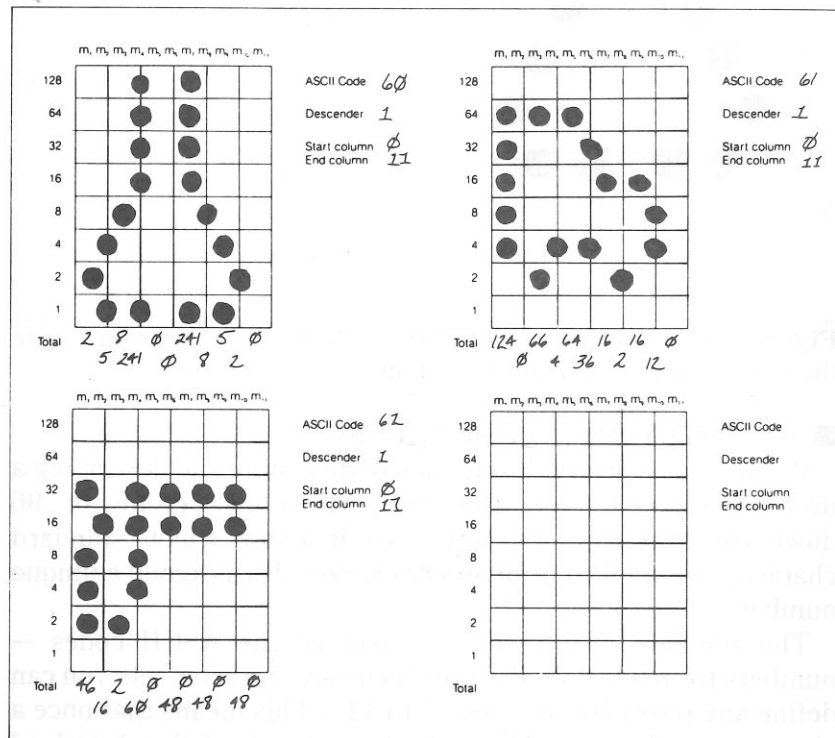


Figure 5-9. Character designs for the three graph symbols.

■ Download character definition command

You've read through a long explanation of download characters and we haven't even told you the command syntax yet! Now the wait is over. This is the most complex command in your printer repertoire and now you've got the necessary knowledge to implement it. Here it is:

```
<ESC>"&"CHR$(0) n1 n2 m0 m1 m2 m3 m4 m5 m6 m7 m8 m9
m10 m11
```

Like the other printer's commands, it starts with an <ESC> (CHR\$(27)). The next character is an ampersand (&) (CHR\$(38)) followed by a CHR\$(0).

n1 and *n2* are used to specify the ASCII values of the characters you are defining. The reason that there are two bytes reserved for this is that your printer allows you to define many characters with just a single command. *n1* is used to specify the beginning of a range of characters to be defined; *n2* specifies the end of the range. For instance, if you wanted to change the appearance of the numerals from 0 to 9 (which have ASCII codes 48 through 57), the command would begin with <ESC> "&" CHR\$(0) CHR\$(48) CHR\$(57) ... Of course, you can also define individual characters by making *n1* and *n2* equal.

m0 is called the attribute byte, for it describes two attributes of the character we have designed: descender data and proportional width information. A byte consists of eight bits. In the attribute byte, the first (high order) bit is used for the descender data, and the last seven bits are used for proportional widths. We'll be discussing proportional character widths in detail later in this chapter; for now, we'll leave it at 11. The descender data was discussed earlier: to use the top eight pins, this bit should be 1; to use the bottom eight pins this bit should be 0. Figure 5-10 shows the bits of the attribute byte as we'll use them for our flask character. By now you've probably seen an easier way to determine the value of the attribute byte. Instead of translating everything to binary, merely assign the descender data a value of 128 (the value of the first bit) if you *don't want* descenders, or 0 if you *want* descenders. Then just add the descender data to the proportional width. This way, it's simply a matter of adding two decimal numbers. (In our case, it's $128 + 11 = 139$.)

0	000	1011	= 11 (decimal)
Descender data	Starting print column	Ending print column	

Figure 5-10. The attribute byte (*m0*) for our flask character.

You'll probably recognize *m1* ... *m11* from the top of our layout grid. That's right, each column is described by one byte. Now we've got everything we need to download one character to the printer. The complete command for our flask character is shown below:

```
CHR$(27);CHR$(38);CHR$(0);CHR$(60);CHR$(60);
CHR$(139);CHR$(2);CHR$(5);CHR$(8);CHR$(241);CHR$(0);
CHR$(0);CHR$(241);CHR$(8);CHR$(5);CHR$(2);CHR$(0)
```

Now let's send the information to the printer. The following program will send the character definitions for all three characters to the printer. Turn off the printer and set DIP switch 1-1 off. Then turn on the printer. Enter the program and run it.

```
10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(27);"&";CHR$(0);CHR$(60);CHR$(62);
30 FOR N=60 TO 62
40 FOR M=0 TO 11
50 READ MM
60 IF PEEK(49601)>127 THEN 60
70 POKE 49296,MM
80 NEXT M
90 NEXT N
100 PRINT
110 DATA 139, 2, 5, 8,241, 0, 0,241, 8,
      5, 2, 0
120 DATA 139,124, 0, 66, 4, 64, 36, 16, 2,
      16, 12, 0
130 DATA 139, 46, 16, 2, 60, 0, 48, 0, 48,
      0, 48, 0
140 PRINT CHR$(4) "PR#0"
```

When you run this program, it looks like nothing happens. That's OK. We'll see why in just a moment. Save this program.

We'll need it again shortly.

PRINTING DOWNLOAD CHARACTERS

You've now defined and sent three characters to your printer. But how do you know that? If you try printing those characters now you don't get a flask, car and gun. Instead you get .. (<=). That's because the download characters are stored in a different part of the printer's memory. To tell it to look in download character RAM instead of standard character ROM it requires another command:

```
<ESC> "% " CHR$(n);CHR$(0)
```

This command is used to select the download character set (if *n*=49) or to select the standard character set (if *n*=48). Let's try it out. Enter this program:

```
10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(27);"%1";CHR$(0);
30 PRINT CHR$(60);CHR$(61);CHR$(62);
40 PRINT CHR$(27);"%0";CHR$(0);
50 PRINT CHR$(4) "PR#0"
```

Voila! It should have printed out the three characters we defined. Your printout should look like this:

```
PR#1
```

(If it doesn't, check the last program we ran for errors, then rerun it.)

Let's find out if there are any other characters in the download RAM. Try this program:

```
10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(27);"%1";CHR$(0);
30 FOR I=32 TO 126
40 PRINT CHR$(I);
50 NEXT I
60 PRINT
```

```
70 PRINT CHR$(27);"%0";CHR$(0)
80 PRINT CHR$(4) "PR#0"
```

Nope! Just three characters in the download set. This is inconvenient for a couple of reasons. First, every time you wanted to use a download character you would have to switch back and forth between character sets. Knowing that you wouldn't want to do that, your printer won't even allow it. So we have made it an easy task to use mostly standard characters with just a few special characters thrown in. This command copies all the ASCII characters from the standard character ROM into download RAM:

```
<ESC> ":" CHR$(0);CHR$(0);CHR$(0)
```

Since it will copy *all* characters into download area, it will wipe out any characters that are already there. So it's important to send this command to the printer before you send any download characters you want to define. With that in mind, add this line to the program we used to send the characters to your printer:

```
15 PRINT CHR$(27);":";CHR$(0);CHR$(0);CHR$(0);
```

Now try the download printout test program again. Your results look like Figure 5-11.

```
!"#$%&'()*+,-./0123456789:;@?`~!ABCDEF GHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
pqrstuvwxyz{|}~"
```

Figure 5-11. Printout of the download character set, into which all the ASCII characters have been copied, and the <, = and > have been changed.

To demonstrate how to use these characters, let's use this character set to print a small graph. This program, which has been built around the first program in this chapter, will do just that:

```
10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(27);":";CHR$(0);CHR$(0);CHR$(0);
30 PRINT CHR$(27);"&";CHR$(0);CHR$(60);CHR$(62);
```

```
40 FOR N=60 TO 62
50 FOR M=0 TO 11
60 READ MM
70 IF PEEK(49601)>127 THEN 70
80 POKE 49296,MM
90 NEXT M
100 NEXT N
110 PRINT
120 DATA 139, 2, 5, 8,241, 0, 0,241, 8,
5, 2, 0
130 DATA 139,124, 0, 66, 4, 64, 36, 16, 2,
16, 12, 0
140 DATA 139, 46, 16, 2, 60, 0, 48, 0, 48,
0, 48, 0
150 PRINT CHR$(27);"(011.";
160 PRINT CHR$(27);"h";CHR$(1);
170 PRINT " U.S. EXPORTS"
180 PRINT CHR$(27);"h";CHR$(0);
190 PRINT CHR$(27);"%1";CHR$(0);
200 PRINT "AUTOS"; :GOSUB 510
210 FOR I=0.4 TO 9.3 STEP 0.4
220 PRINT CHR$(61);
230 NEXT I
240 PRINT
250 PRINT "CHEMICALS"; :GOSUB 510
260 FOR I=0.4 TO 8.7 STEP 0.4
270 PRINT CHR$(60);
280 NEXT I
290 PRINT
300 PRINT "GUNS"; :GOSUB 510
310 FOR I=0.4 TO 1.4 STEP 0.4
320 PRINT CHR$(62);
330 NEXT I
340 PRINT
350 GOSUB 510 :PRINT "+--";
360 S$="--+--"
370 FOR I=2 TO 8 STEP 2
380 PRINT S$;
390 NEXT I
400 PRINT "--+"
410 GOSUB 510 :PRINT " ";
420 FOR I=2 TO 8 STEP 2
430 PRINT " ";I;
440 NEXT I
450 PRINT CHR$(27);"%0";CHR$(0)
460 PRINT CHR$(27);"x";
470 GOSUB 510 :PRINT "MILLIONS OF DOLLARS"
```


As you've learned in Chapter 2, you can print NLQ characters. You can also define the download characters with NLQ mode. Since NLQ characters use many more dots than draft characters, defining NLQ download characters is more complex than designing draft ones. If you use the grid and the program in this section, however, you will be able to design your own NLQ characters.

Because the NLQ characters can use as many as 16 dots vertically and 23 dots horizontally, you plan your designs on a different grid than the one you used for draft characters. Make up some grids (photocopy Figure 5-12 if you wish) and get ready to be creative!

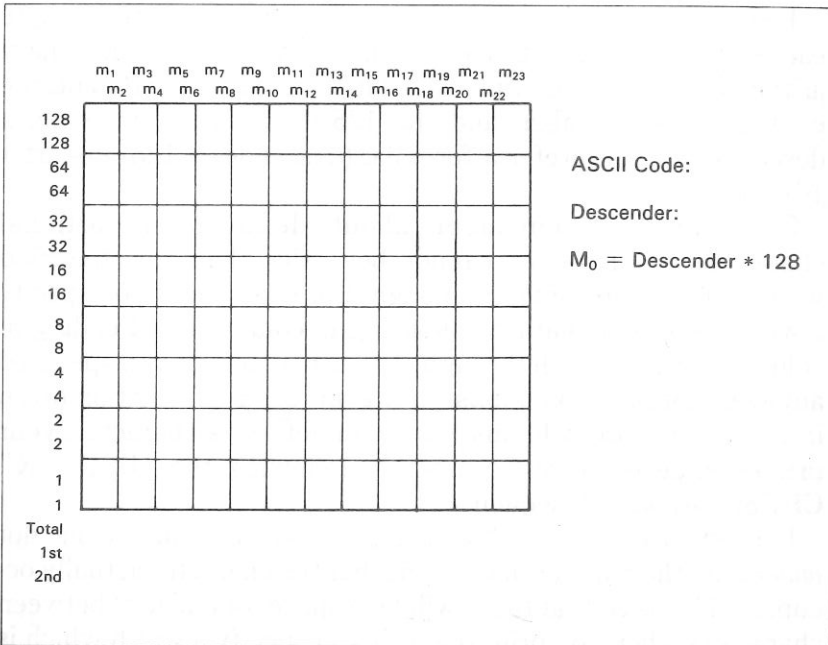


Figure 5-12. Use this grid (or one similar to it) to define your own NLQ characters.

As you noticed when the NLQ characters are printed, they are printing in twice; the first line of data is printed, the paper is moved up a distance of 1/2 dot, then the second data line is printed. So, we've written the numbers on the horizontal lines.

To calculate the data numbers for this column, you see which dots are used in the box and add their values together. Then you go down the dots on the horizontal lines and add their values together as shown in Figure 5-13.

Now we'll show you how to use the NLQ character definition with a flask as shown in Figure 5-13. Figure 5-13 shows the design drawn on a grid and the data numbers printed at the bottom of each column.

If you look at each column individually, you can see how the data numbers were calculated.

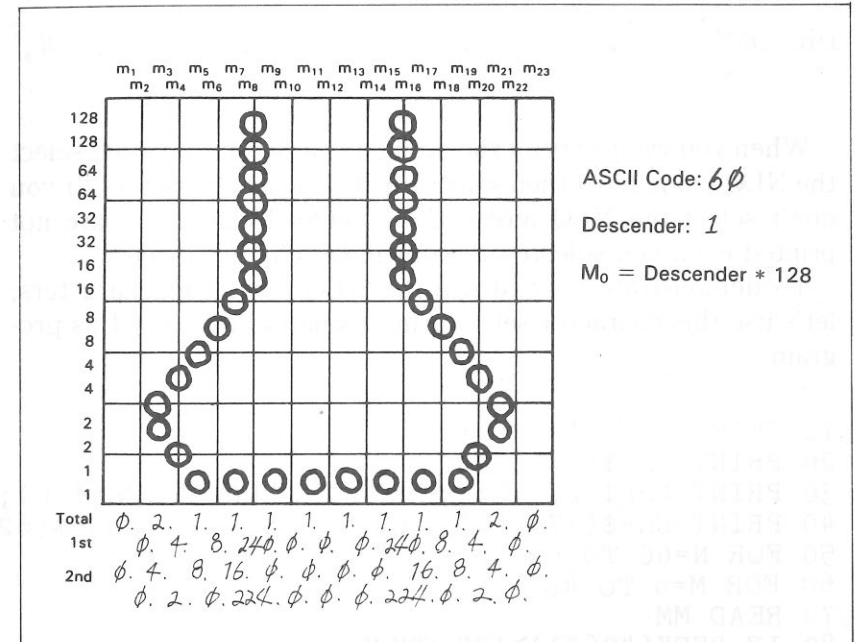


Figure 5-13. Add the values of the dots in each box and line column and write the sum of each column at the bottom.

Now enter the following program and run it. It has the data numbers for the NLQ flask character. For a character of your own, change the DATA numbers and the character definition position.

```

10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(27); "a2";
30 PRINT CHR$(27); "&"; CHR$(0); CHR$(60); CHR$(60);
40 FOR M=0 TO 46
50 READ MM
60 IF PEEK(49601)>127 THEN 60
70 POKE 49296,MM
80 NEXT M
90 PRINT
    
```

```

100 PRINT CHR$(27);"a0"
110 PRINT CHR$(4) "PR#0"
120 END
130 DATA 128, 0, 0, 2, 4, 1, 8, 1,240,
1, 0, 1
140 DATA 0, 1, 0, 1,240, 1, 8, 1, 4,
2, 0, 0
150 DATA 0, 0, 4, 2, 8, 0, 16,224, 0,
0, 0, 0
160 DATA 0, 0, 0,224, 16, 0, 8, 2, 4,
0, 0

```

When you want to print the defined character, you must select the NLQ mode first, then select the download characters. If you don't select the NLQ mode, the download characters are not printed even you selected the download character set.

To demonstrate how to use the NLQ download characters, let's use this character set to print a small graph. Try this program.

```

10 PRINT CHR$(4) "PR#1"
20 PRINT CHR$(27);"a2";
30 PRINT CHR$(27);":":CHR$(0);CHR$(0);CHR$(0);
40 PRINT CHR$(27);"&":CHR$(0);CHR$(60);CHR$(62);
50 FOR N=60 TO 62
60 FOR M=0 TO 46
70 READ MM
80 IF PEEK(49601)>127 THEN 80
90 POKE 49296,MM
100 NEXT M
110 NEXT N
120 PRINT
130 DATA 128, 0, 0, 2, 4, 1, 8, 1,240,
1, 0, 1
140 DATA 0, 1, 0, 1,240, 1, 8, 1, 4,
2, 0, 0
150 DATA 0, 0, 4, 2, 8, 0, 16,224, 0,
0, 0, 0
160 DATA 0, 0, 0,224, 16, 0, 8, 2, 4,
0, 0
170 DATA 128, 14, 16, 38, 1, 70, 1, 70, 0,
126, 0, 0
180 DATA 70, 0, 38, 0, 22, 9, 6, 9, 6,
8, 6, 0
190 DATA 28, 32, 14, 64, 14, 0, 14, 0,124,
0, 0, 12

```

```

200 DATA 64, 12, 32, 14, 16, 14, 0, 14, 0,
12, 0
210 DATA 128, 0, 24, 7, 56, 1, 56, 7, 8,
2, 8, 6
220 DATA 8, 2, 8, 4, 8, 0, 8, 32, 8,
0, 24, 0
230 DATA 32, 16, 14, 48, 14, 48, 14, 32, 0,
32, 0, 40
240 DATA 0, 36, 0, 40, 0, 32, 0, 32, 0,
48, 0
250 PRINT CHR$(27);"(011.";
260 PRINT CHR$(27);"h";CHR$(1);
270 PRINT " U.S. EXPORTS"
280 PRINT CHR$(27);"h";CHR$(0);
290 PRINT CHR$(27);"%1";CHR$(0);
300 PRINT "AUTOS";CHR$(9);
310 FOR I=0.4 TO 9.3 STEP 0.4
320 PRINT CHR$(61);
330 NEXT I
340 PRINT
350 PRINT "CHEMICALS";CHR$(9);
360 FOR I=0.4 TO 8.7 STEP 0.4
370 PRINT CHR$(60);
380 NEXT I
390 PRINT
400 PRINT "GUNS";CHR$(9);
410 FOR I=0.4 TO 1.4 STEP 0.4
420 PRINT CHR$(62);
430 NEXT I
440 PRINT
450 PRINT CHR$(9);"+--";
460 S$="--+--"
470 FOR I=2 TO 8 STEP 2
480 PRINT S$;
490 NEXT I
500 PRINT "--+"
510 PRINT CHR$(9);" ";
520 FOR I=2 TO 8 STEP 2
530 PRINT " ";I;
540 NEXT I
550 PRINT CHR$(27);"%0";CHR$(0)
560 PRINT CHR$(27);"a0";
570 PRINT CHR$(27);"x";
580 PRINT CHR$(9);"MILLIONS OF DOLLARS"
590 PRINT CHR$(27);"z"
600 PRINT CHR$(4) "PR#0"
610 END

```


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 11 dots or higher than 8 dots.
- If an image is to be printed just one time, as opposed to a frequently used “text” character.

USING THE DOT GRAPHICS COMMANDS

The command to print dot graphics uses this format:

<ESC> “G” d_4 $m1$ $m2$...

Just like many of the other codes you have learned, the command starts with an escape sequence (<ESC> “G” in this case). But unlike the other codes there can be any number of graphics data bytes following the command. That’s where d_4 come in; they are used to tell the printer how many bytes of graphics data to expect with four-digit numbers.

To figure the values of d_4 , you’ll need to figure out how wide your graphics image will be.

■ 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 the firing of every single pin on the print head. In Figure 6-1, 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$(2)`. Firing several pins at once is done in a similar fashion. For example, to print the first, third, and fourth dots, add their values (1 + 4 + 8) to send this total: `CHR$(13)`. This is one byte of graphics data; it would replace $m1$ in our format statement.

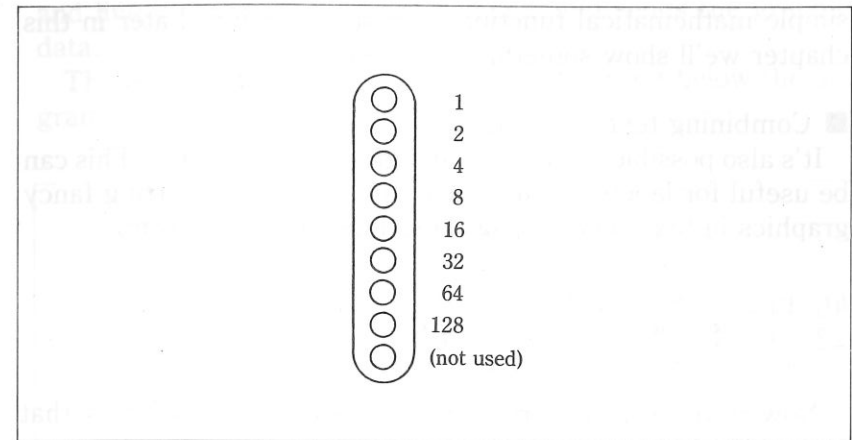


Figure 6-1. Starting with the least significant bit at the top, each pin of the print head is assigned a value which is a power of two.

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

~~~~~

```

10 REM Demo of dot graphics
20 PRINT CHR$(4) "PR#1"
30 PRINT
40 PRINT CHR$(27);"N";
50 PRINT CHR$(27);"G";"0350";
60 FOR I=1 TO 25
70 FOR J=6 TO 0 STEP -1
80 PRINT CHR$(2^J);
90 NEXT J
100 FOR J=0 TO 6
110 PRINT CHR$(2^J);
120 NEXT J
130 NEXT I
140 PRINT
150 PRINT CHR$(4) "PR#0"

```

In lines 40 and 50, we’ve selected normal density graphics and said that 350 characters of graphics data would follow. The loop between lines 60 and 130 is repeated 25 times; this is what gives us the “zigzag” effect. The loop from line 70 to line 90 creates the lines that slope up; the loop between lines 100 and 120 prints the downward sloping lines. This is an example of using a very



```

230 DATA 240,240, 0, 0, 0, 0, 0, 0, 0, 0
240 DATA 0,128,192,192,224,224,240,112,112,112
250 DATA 112,240,224,224,224,192, 0, 0, 0, 0
260 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
270 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
280 DATA 0, 0, 0, 0,128,192,224,224,224,240
290 DATA 112,112,112,112,112,224,224,192,192,240
300 DATA 240,240, 0, 0, 0, 0, 0, 0, 0, 0
310 REM ROW 2
320 DATA 0, 0, 60,255,255,255,255,255,241,240
330 DATA 224,224,224,224,192,192,192,193,131,143
340 DATA 15, 15, 0, 0, 0, 0, 0, 0, 0,128
350 DATA 158,191,191,255,255,255,241,224,224,224
360 DATA 248,192, 63, 31, 31, 15, 3, 0,224,240
370 DATA 248,248,240,224,192, 0, 0, 0, 0, 0
380 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
390 DATA 0, 0, 60,255,255,255,255,255,241,240
400 DATA 224,224,224,224,192,192,192,193,131,143
410 DATA 15, 15, 0, 0, 0, 0, 0, 0, 0, 0
420 REM ROW 3
430 DATA 0,248,248,192,129, 1, 3, 3, 3, 3
440 DATA 3, 7, 7, 7, 7, 15,255,255,255,255
450 DATA 255,254, 0, 0, 0, 0,252,254,255,255
460 DATA 255,255,131, 1, 1, 1, 1, 3, 7, 15
470 DATA 63,255,255,255,254,252,248,224,224,248
480 DATA 127, 63, 31, 7, 1, 0, 0,192,224,224
490 DATA 224,192, 0, 0, 0, 0, 0, 0, 0, 0
500 DATA 0,248,248,192,129, 1, 3, 3, 3, 3
510 DATA 3, 7, 7, 7, 7, 15,255,255,255,255
520 DATA 255,254, 0, 0, 0, 0, 0, 0, 0, 0
530 REM ROW 4
540 DATA 0, 31, 31, 15, 7, 7, 14, 14, 28, 28
550 DATA 28, 28, 28, 30, 30, 15, 15, 7, 7, 3
560 DATA 1, 0, 0, 0, 0, 0, 3, 7, 15, 15
570 DATA 15, 31, 31, 31, 30, 30, 28, 28, 28, 28
580 DATA 12, 14, 7, 7, 7, 7, 15, 15, 31, 31
590 DATA 30, 30, 28, 28, 28, 28, 30, 15, 7, 7
600 DATA 3, 1, 0, 0, 0, 0, 0, 0, 0, 0
610 DATA 0, 31, 31, 15, 7, 7, 14, 14, 28, 28
620 DATA 28, 28, 28, 30, 30, 15, 15, 7, 7, 3
630 DATA 1, 0, 0, 0, 0, 0, 0, 0, 0, 0

```

**S&S**

## 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 80 dots per inch horizontally and 72 dots per inch vertically, it takes at least 720 bytes of memory for each square inch of plotted area. That doesn't sound so bad — but an area 8 inches square requires over 32K!)

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).

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.

```

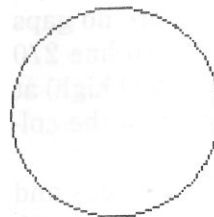
10 REM Set program constants
20 A=24576
30 FOR I=A TO A+12
40 READ B
50 POKE I,B
60 NEXT I
70 DATA 32, 74,255,165,250, 5,251
80 DATA 133,252, 32, 63,255, 96
90 DIM BIT%(100,14)
100 MASK%(1)=2 :MASK%(4)=16
110 MASK%(2)=4 :MASK%(5)=32
120 MASK%(3)=8 :MASK%(6)=64
130 LX=20 :LY=20

```

```

140 XFAC=96/LX      :YFAC=87/LY
150 REM Plot curve
160 GOSUB 600
170 REM Send bit image map to printer
180 PRINT CHR$(4) "PR#1"
190 PRINT CHR$(9) "ON"
200 PRINT CHR$(27);"T12"
210 FOR ROW=0 TO 14
220 A$=""
230 FOR COL=1 TO 100
240 A$=A$+CHR$(BIT%(COL,ROW))
250 NEXT COL
260 PRINT CHR$(27);"G0100";A$
270 NEXT ROW
280 PRINT CHR$(4) "PR#0"
290 END
300 REM Subroutine to draw a line from X1,Y1
    to X2,Y2
310 XL=X2-X1      :YL=Y2-Y1
320 NX=ABS(XL*XFAC):NY=ABS(YL*YFAC)
330 IF NX<NY THEN NX=NY
340 NS%=INT(NX+1)
350 DX=XL/NS%    :DY=YL/NS%
360 FOR I=1 TO NS%
370 X1=X1+DX     :Y1=Y1+DY
380 GOSUB 500
390 NEXT I
400 RETURN
500 REM Subroutine to plot a point at X1,Y1
510 XX=X1*XFAC   :YY=Y1*YFAC
520 COL=INT(XX)+1
530 ROW=INT(YY/6)
540 XIT%=INT(YY-(6*ROW))+1
550 POKE 250,BIT%(COL,ROW)
560 POKE 251,MASK%(XIT%)
570 CALL 24576
580 BIT%(COL,ROW)=PEEK(252)
590 RETURN
600 REM Subroutine to plot a circle
610 RAD=9
620 X1=19       :Y1=10
630 FOR ANG=0 TO 360 STEP 10
640 R1=ANG*6.28/360
650 X2=RAD*COS(R1)+10
660 Y2=RAD*SIN(R1)+10
670 GOSUB 300
680 NEXT ANG
690 RETURN

```



#### ■ How the program works

In the program above, we've created an array called BIT%, which is dimensioned in line 90. The array MASK% contains the values of the dots. (In order to make this program run on most Apple computers, we're only six pins for graphics. With many computers, you can use all eight available pins.) In lines 130 and 140 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 300. This subroutine calculates the coordinates of individual points along that line.

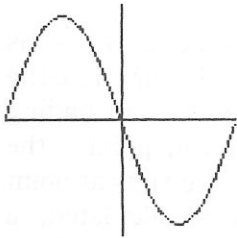
After these coordinates have been determined, the subroutine at line 500 is called. This routine turns "on" an individual dot in our array called BIT%. (Keep in mind that no printing has been done yet; the computer is still drawing the image on its "graph paper" in memory.) The way an individual dot is turned on is using the POKE function in lines 540 to 580.

When all the points have been plotted in memory, printing begins at line 180. We first set the line spacing to 12/144 inch us-

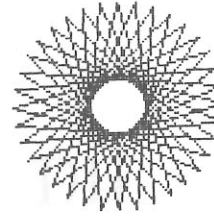
ing the  $\langle \text{ESC} \rangle$  "T" command. This is so that there are no gaps between rows of dots. Then the loop from line 210 to line 270 prints the dot graphics image one line (which is six dots high) at a time. The variable A\$ is used to build a string of all the columns of BIT% in a given row.

As you can see, by taking the program in small pieces and analyzing it, graphics 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 REM Subroutine to plot a sine wave
610 X1=0      :Y1=10
620 X2=20    :Y2=10
630 GOSUB 300
640 X1=10    :Y1=0
650 X2=10    :Y2=20
660 GOSUB 300
670 X1=0     :Y1=10
680 FOR X2=0 TO 20 STEP 0.2
690 Y2=10-9*SIN(3.14159*X2/10)
700 GOSUB 300
710 NEXT X2
720 RETURN
```



```
600 REM Subroutine to plot a star
610 RAD=9
620 FOR ANG=0 TO 360 STEP 10
630 R1=ANG*6.28/360
640 R2=(ANG+150)*6.28/360
650 X1=RAD*COS(R1)+10
660 Y1=RAD*SIN(R1)+10
670 X2=RAD*COS(R2)+10
680 Y2=RAD*SIN(R2)+10
690 GOSUB 300
700 NEXT ANG
710 RETURN
```



## HIGH RESOLUTION GRAPHICS

Up until now all of the dot graphics printing we have done has been with your printer's normal density mode. 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 five density modes you can use; they're summarized in Table 6-1.

**Table 6-1**  
Dot graphics density selection commands

| Function                   | Control code                                                                                             |
|----------------------------|----------------------------------------------------------------------------------------------------------|
| 72 dots/inch density mode  | $\langle \text{ESC} \rangle$ "n"                                                                         |
| 80 dots/inch density mode  | $\langle \text{ESC} \rangle$ "N" or $\langle \text{ESC} \rangle$ "a" 2                                   |
| 90 dots/inch density mode  | $\langle \text{ESC} \rangle$ "E"                                                                         |
| 120 dots/inch density mode | $\langle \text{ESC} \rangle$ "e" or $\langle \text{ESC} \rangle$ "q"                                     |
| 160 dots/inch density mode | $\langle \text{ESC} \rangle$ "P" or $\langle \text{ESC} \rangle$ "p" or $\langle \text{ESC} \rangle$ "Q" |

The density of dot graphics is determined by the print pitch just before the dot graphics mode is selected.

Using the first program in this chapter, change line 40 to try each of the different modes. Just change "N" to "n", "E", "q", or "Q" in turn. Your printouts should look something like this:

72 dots/inch density mode with  $\langle \text{ESC} \rangle$  "n"



80 dots/inch density mode with  $\langle \text{ESC} \rangle$  "N"



90 dots/inch density mode with  $\langle \text{ESC} \rangle$  "E"



120 dots/inch density mode with  $\langle \text{ESC} \rangle$  "e"



160 dots/inch density mode with  $\langle \text{ESC} \rangle$  "Q"

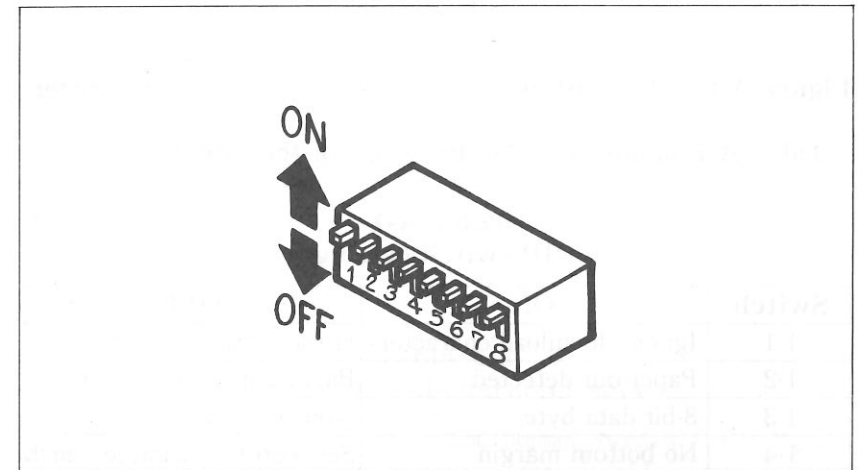


As you can see, the different modes seem to condense the printed image. So, to get the same image in a higher density mode, you must plot more points. This requires twice as much memory for your array, twice as much computing time, and twice as much printing time (but the results may be worth it!).

## APPENDIX A

# DIP SWITCH SETTINGS

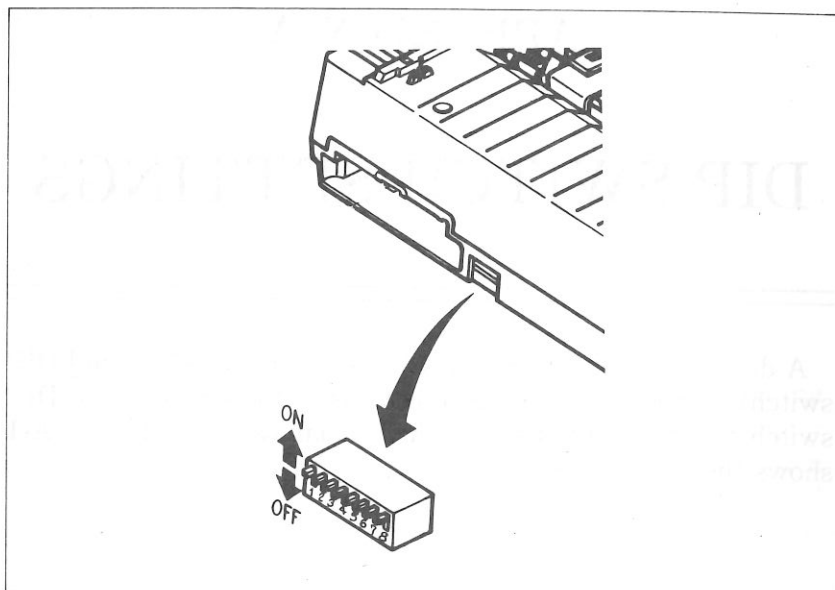
A dual-in-line set of switches (collectively called a [one] DIP switch) controls some of the functions of the printer. The DIP switch actually contains eight individual switches. Figure A-1 shows the DIP switch.



**Figure A-1.** The DIP switch is several small switches in one package.

The DIP switch is readily accessible from the rear of the printer. To set one of the switches, use a ball-point pen to move the switch lever gently. The on position is upwards, and off is downwards.

**Warning:** Never set a DIP switch when the power is on. Turn off both the printer and your computer first.



**Figure A-2.** The DIP switch is located on the back of the printer.

Table A-1 summarizes the functions of the switches.

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

| Switch | ON                                                     | OFF                         |
|--------|--------------------------------------------------------|-----------------------------|
| 1-1    | Ignore download characters                             | Enable download characters  |
| 1-2    | Paper-out detected                                     | Paper-out not detected      |
| 1-3    | 8-bit data byte                                        | 7-bit data byte             |
| 1-4    | No bottom margin                                       | Set bottom margin to 1 inch |
| 1-5    | 11" page length                                        | 12" page length             |
| 1-6    | International character set selection — see Table A-2. |                             |
| 1-7    |                                                        |                             |
| 1-8    |                                                        |                             |

## SWITCH FUNCTIONS

| Switch | Function                                                                                                                                                                                                |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1-1    | 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.

- 1-2 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.
- 1-3 This switch set the bit length for a data byte. When this switch is on, the data byte is set to 8-bit length. When this switch is off, the data byte is set to 7-bit length. When the data byte is set to 7-bit, the "high-order" characters cannot be accepted.
- 1-4 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. You can set the bottom margin with control code.
- 1-5 This switch sets the default page length. When the switch is on, the page length is 11 inches. When the switch is off, the page length is 12 inches.
- 1-6~1-8 These switches determine the default international character set as shown in Table A-2. You can select all international character sets with the control codes.

**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   |



## MEMO

---



---

## APPENDIX B

# ASCII CODES AND CONVERSION CHART

---



---

| Standard ASCII Codes |      |           | Control   | Character set |       |
|----------------------|------|-----------|-----------|---------------|-------|
| Decimal              | Hex. | Binary    | Character | Draft         | NLQ   |
| 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    |               |       |
| 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    |               |       |
| 18                   | 12   | 0001 0010 | Ctrl-R    |               |       |
| 19                   | 13   | 0001 0011 | Ctrl-S    |               |       |
| 20                   | 14   | 0001 0100 | Ctrl-T    |               |       |
| 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 |           | GS            | GS    |
| 30                   | 1E   | 0001 1110 |           | RS            | RS    |
| 31                   | 1F   | 0001 1111 |           | US            | US    |
| 32                   | 20   | 0010 0000 |           | Space         | Space |

| Standard ASCII Codes |             |           | Character set |       |
|----------------------|-------------|-----------|---------------|-------|
| Decimal              | Hexadecimal | Binary    | Draft         | NLQ   |
| 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    | Draft         | NLQ |
| 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    | Draft         | NLQ |
| 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 |               |     |
| 128                  | 80          | 1000 0000 |               |     |
| 129                  | 81          | 1000 0001 |               |     |
| 130                  | 82          | 1000 0010 |               |     |
| 131                  | 83          | 1000 0011 |               |     |
| 132                  | 84          | 1000 0100 |               |     |
| 133                  | 85          | 1000 0101 |               |     |
| 134                  | 86          | 1000 0110 |               |     |
| 135                  | 87          | 1000 0111 |               |     |
| 136                  | 88          | 1000 1000 |               |     |
| 137                  | 89          | 1000 1001 |               |     |
| 138                  | 8A          | 1000 1010 |               |     |
| 139                  | 8B          | 1000 1011 |               |     |
| 140                  | 8C          | 1000 1100 |               |     |
| 141                  | 8D          | 1000 1101 |               |     |
| 142                  | 8E          | 1000 1110 |               |     |
| 143                  | 8F          | 1000 1111 |               |     |
| 144                  | 90          | 1001 0000 |               |     |
| 145                  | 91          | 1001 0001 |               |     |
| 146                  | 92          | 1001 0010 |               |     |
| 147                  | 93          | 1001 0011 |               |     |
| 148                  | 94          | 1001 0100 |               |     |
| 149                  | 95          | 1001 0101 |               |     |
| 150                  | 96          | 1001 0110 |               |     |
| 151                  | 97          | 1001 0111 |               |     |
| 152                  | 98          | 1001 1000 |               |     |
| 153                  | 99          | 1001 1001 |               |     |
| 154                  | 9A          | 1001 1010 |               |     |
| 155                  | 9B          | 1001 1011 |               |     |
| 156                  | 9C          | 1001 1100 |               |     |
| 157                  | 9D          | 1001 1101 |               |     |
| 158                  | 9E          | 1001 1110 |               |     |
| 159                  | 9F          | 1001 1111 |               |     |
| 160                  | A0          | 1010 0000 |               |     |
| 161                  | A1          | 1010 0001 |               |     |
| 162                  | A2          | 1010 0010 |               |     |
| 163                  | A3          | 1010 0011 |               |     |
| 164                  | A4          | 1010 0100 |               |     |

| Standard ASCII Codes |             |           | Standard ASCII Codes |             |
|----------------------|-------------|-----------|----------------------|-------------|
| Decimal              | Hexadecimal | Binary    | Decimal              | Hexadecimal |
| 165                  | A5          | 1010 0101 | 165                  | A5          |
| 166                  | A6          | 1010 0110 | 166                  | A6          |
| 167                  | A7          | 1010 0111 | 167                  | A7          |
| 168                  | A8          | 1010 1000 | 168                  | A8          |
| 169                  | A9          | 1010 1001 | 169                  | A9          |
| 170                  | AA          | 1010 1010 | 170                  | AA          |
| 171                  | AB          | 1010 1011 | 171                  | AB          |
| 172                  | AC          | 1010 1100 | 172                  | AC          |
| 173                  | AD          | 1010 1101 | 173                  | AD          |
| 174                  | AE          | 1010 1110 | 174                  | AE          |
| 175                  | AF          | 1010 1111 | 175                  | AF          |
| 176                  | B0          | 1011 0000 | 176                  | B0          |
| 177                  | B1          | 1011 0001 | 177                  | B1          |
| 178                  | B2          | 1011 0010 | 178                  | B2          |
| 179                  | B3          | 1011 0011 | 179                  | B3          |
| 180                  | B4          | 1011 0100 | 180                  | B4          |
| 181                  | B5          | 1011 0101 | 181                  | B5          |
| 182                  | B6          | 1011 0110 | 182                  | B6          |
| 183                  | B7          | 1011 0111 | 183                  | B7          |
| 184                  | B8          | 1011 1000 | 184                  | B8          |
| 185                  | B9          | 1011 1001 | 185                  | B9          |
| 186                  | BA          | 1011 1010 | 186                  | BA          |
| 187                  | BB          | 1011 1011 | 187                  | BB          |
| 188                  | BC          | 1011 1100 | 188                  | BC          |
| 189                  | BD          | 1011 1101 | 189                  | BD          |
| 190                  | BE          | 1011 1110 | 190                  | BE          |
| 191                  | BF          | 1011 1111 | 191                  | BF          |
| 192                  | C0          | 1100 0000 | 192                  | C0          |
| 193                  | C1          | 1100 0001 | 193                  | C1          |
| 194                  | C2          | 1100 0010 | 194                  | C2          |
| 195                  | C3          | 1100 0011 | 195                  | C3          |
| 196                  | C4          | 1100 0100 | 196                  | C4          |
| 197                  | C5          | 1100 0101 | 197                  | C5          |
| 198                  | C6          | 1100 0110 | 198                  | C6          |
| 199                  | C7          | 1100 0111 | 199                  | C7          |
| 200                  | C8          | 1100 1000 | 200                  | C8          |
| 201                  | C9          | 1100 1001 | 201                  | C9          |
| 202                  | CA          | 1100 1010 | 202                  | CA          |
| 203                  | CB          | 1100 1011 | 203                  | CB          |
| 204                  | CC          | 1100 1100 | 204                  | CC          |
| 205                  | CD          | 1100 1101 | 205                  | CD          |
| 206                  | CE          | 1100 1110 | 206                  | CE          |
| 207                  | CF          | 1100 1111 | 207                  | CF          |
| 208                  | D0          | 1101 0000 | 208                  | D0          |

Standard ASCII Codes

| Decimal | Hexadecimal | Binary    |
|---------|-------------|-----------|
| 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     | DA          | 1101 1010 |
| 219     | DB          | 1101 1011 |
| 220     | DC          | 1101 1100 |
| 221     | DD          | 1101 1101 |
| 222     | DE          | 1101 1110 |
| 223     | DF          | 1101 1111 |
| 224     | E0          | 1110 0000 |
| 225     | E1          | 1110 0001 |
| 226     | E2          | 1110 0010 |
| 227     | E3          | 1110 0011 |
| 228     | E4          | 1110 0100 |
| 229     | E5          | 1110 0101 |
| 230     | E6          | 1110 0110 |
| 231     | E7          | 1110 0111 |
| 232     | E8          | 1110 1000 |
| 233     | E9          | 1110 1001 |
| 234     | EA          | 1110 1010 |
| 235     | EB          | 1110 1011 |
| 236     | EC          | 1110 1100 |
| 237     | ED          | 1110 1101 |
| 238     | EE          | 1110 1110 |
| 239     | EF          | 1110 1111 |
| 240     | F0          | 1111 0000 |
| 241     | F1          | 1111 0001 |
| 242     | F2          | 1111 0010 |
| 243     | F3          | 1111 0011 |
| 244     | F4          | 1111 0100 |
| 245     | F5          | 1111 0101 |
| 246     | F6          | 1111 0110 |
| 247     | F7          | 1111 0111 |
| 248     | F8          | 1111 1000 |
| 249     | F9          | 1111 1001 |
| 250     | FA          | 1111 1010 |
| 251     | FB          | 1111 1011 |
| 252     | FC          | 1111 1100 |

Standard ASCII Codes

| Decimal | Hexadecimal | Binary    |
|---------|-------------|-----------|
| 253     | FD          | 1111 1101 |
| 254     | FE          | 1111 1110 |
| 255     | FF          | 1111 1111 |

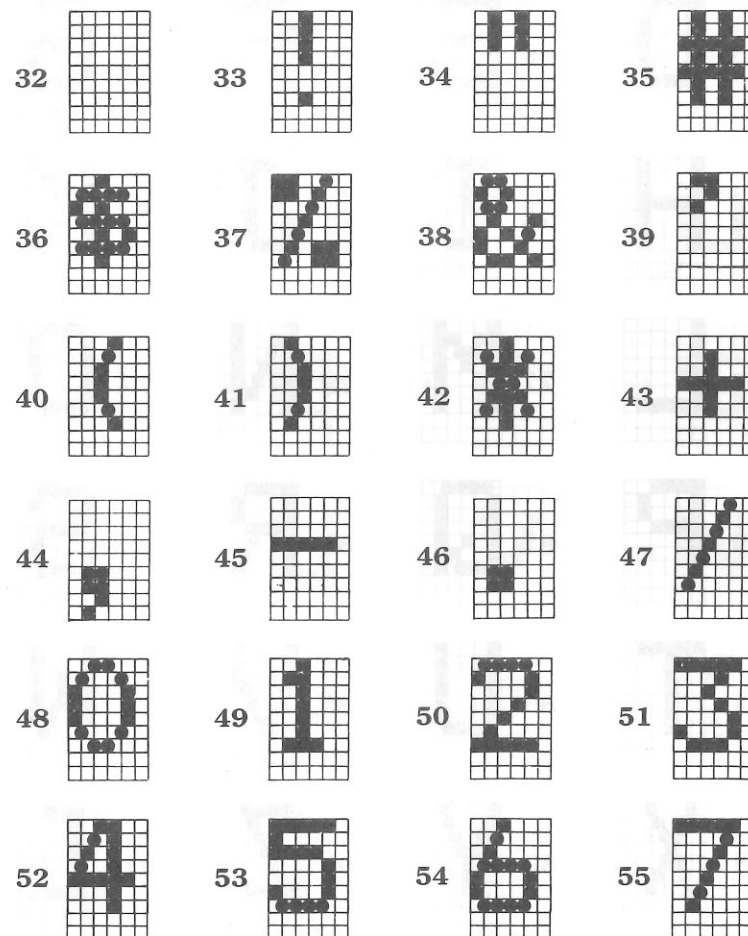
## MEMO

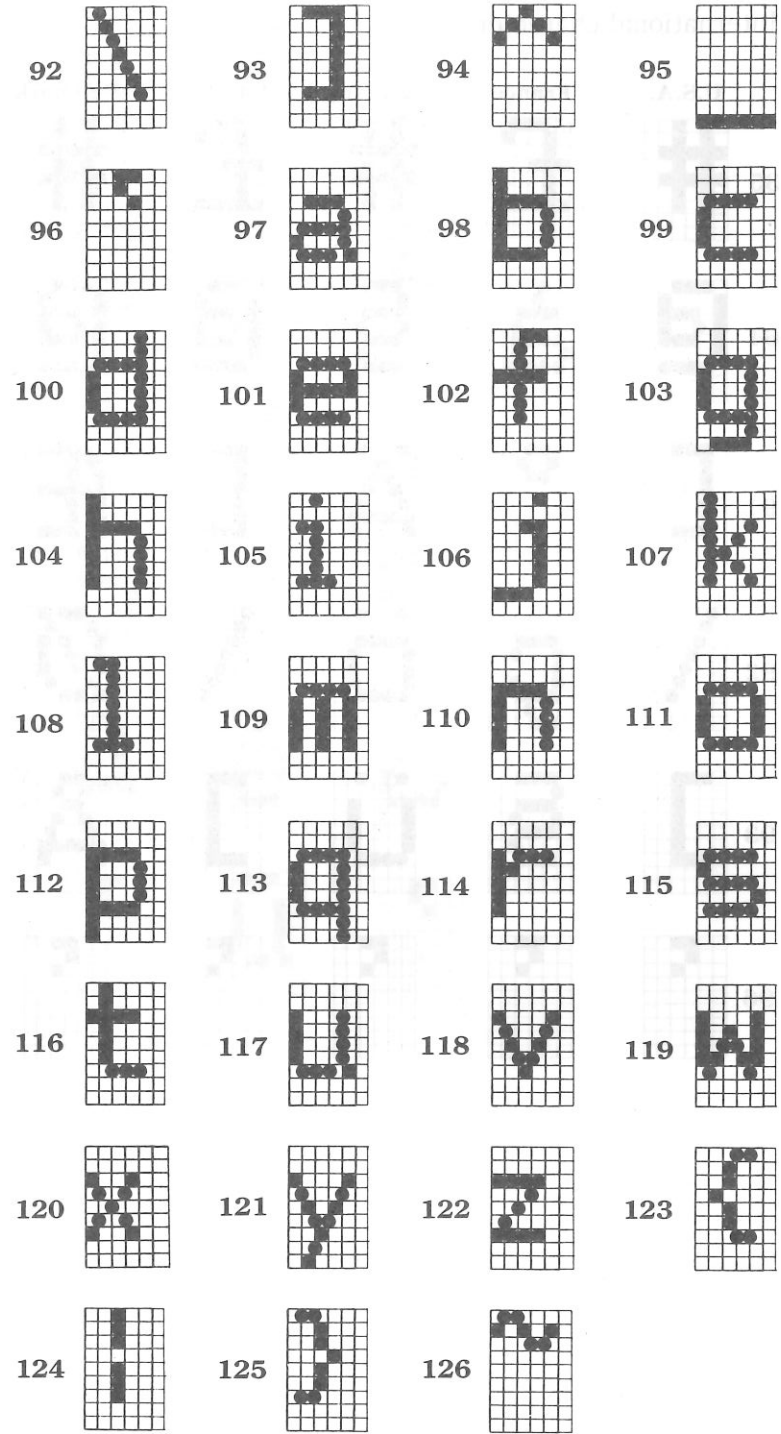
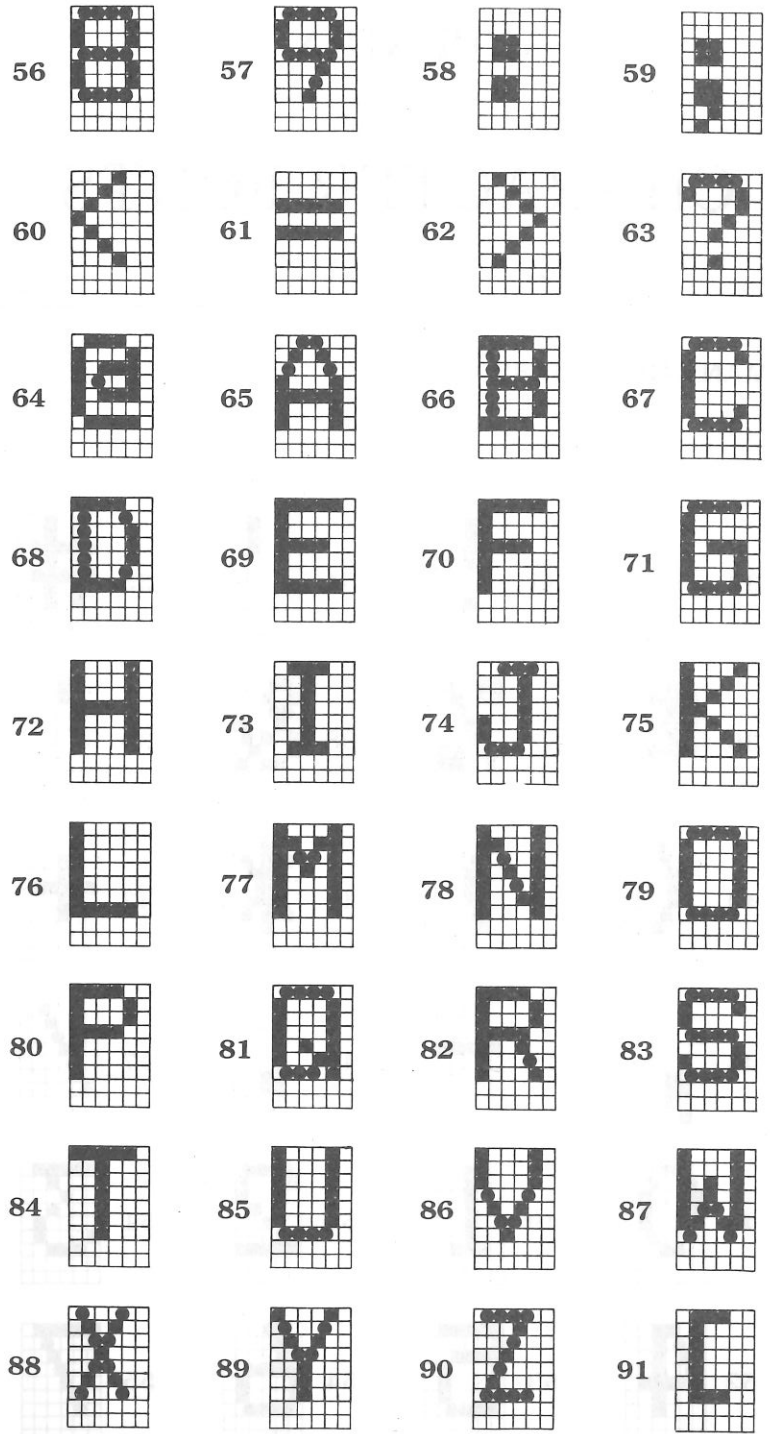
## APPENDIX C

## CHARACTER FONTS

## ROMAN CHARACTERS

## ■ Standard characters





■ International characters

|    | U.S.A. | France | Germany | England | Denmark |
|----|--------|--------|---------|---------|---------|
| 35 |        |        |         |         |         |
| 64 |        |        |         |         |         |
| 91 |        |        |         |         |         |
| 92 |        |        |         |         |         |
| 93 |        |        |         |         |         |
| 96 |        |        |         |         |         |

|    | Sweden | Italy | Spain |
|----|--------|-------|-------|
| 35 |        |       |       |
| 64 |        |       |       |
| 91 |        |       |       |
| 92 |        |       |       |
| 93 |        |       |       |
| 96 |        |       |       |

U.S.A.

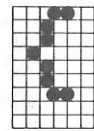
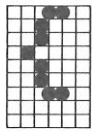
France

Germany

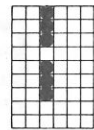
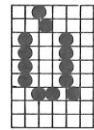
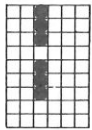
England

Denmrk

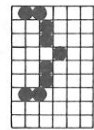
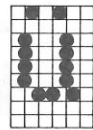
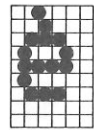
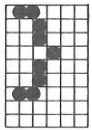
123



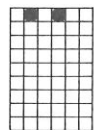
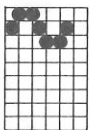
124



125



126

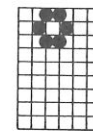


Sweden

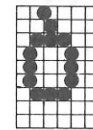
Italy

Spain

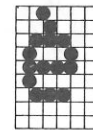
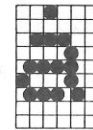
123



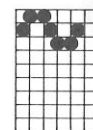
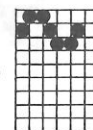
124



125



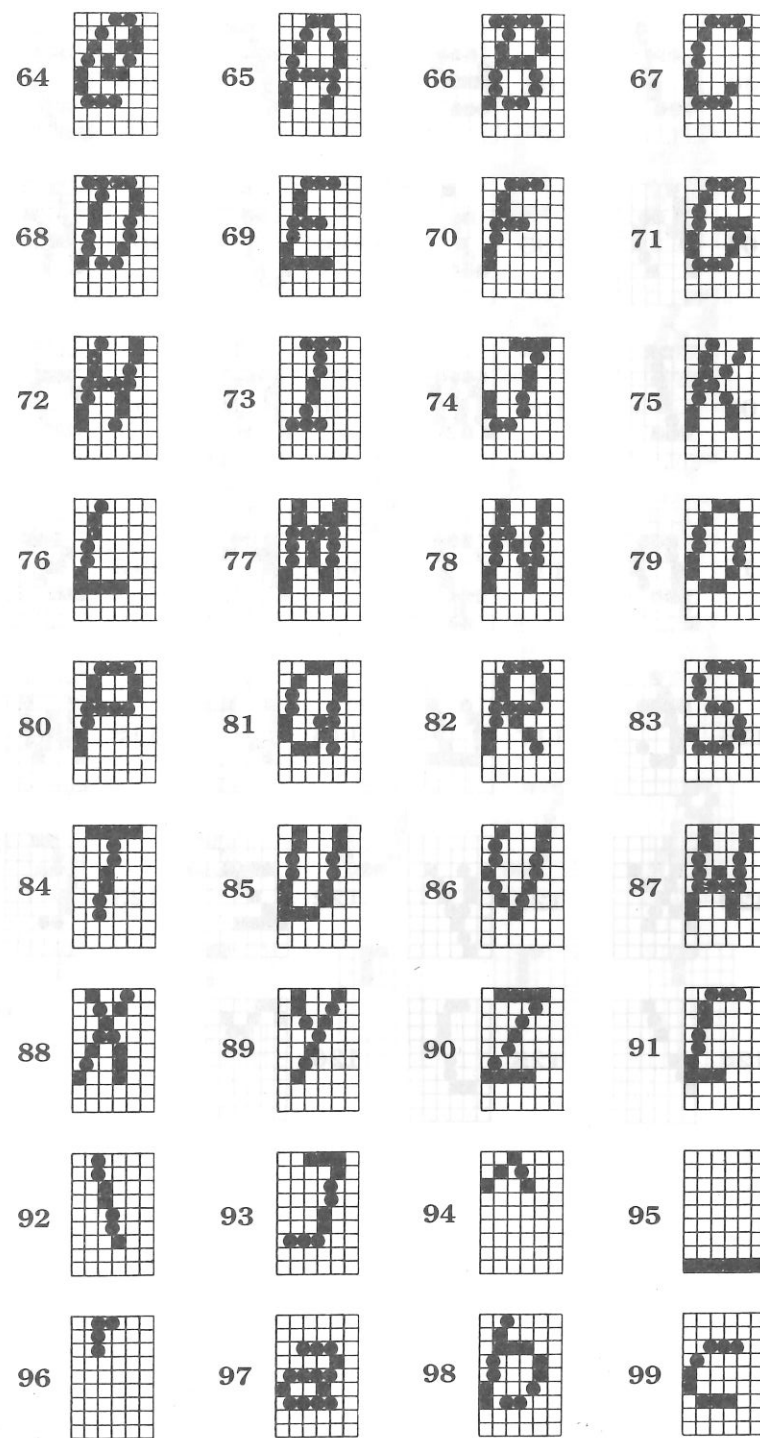
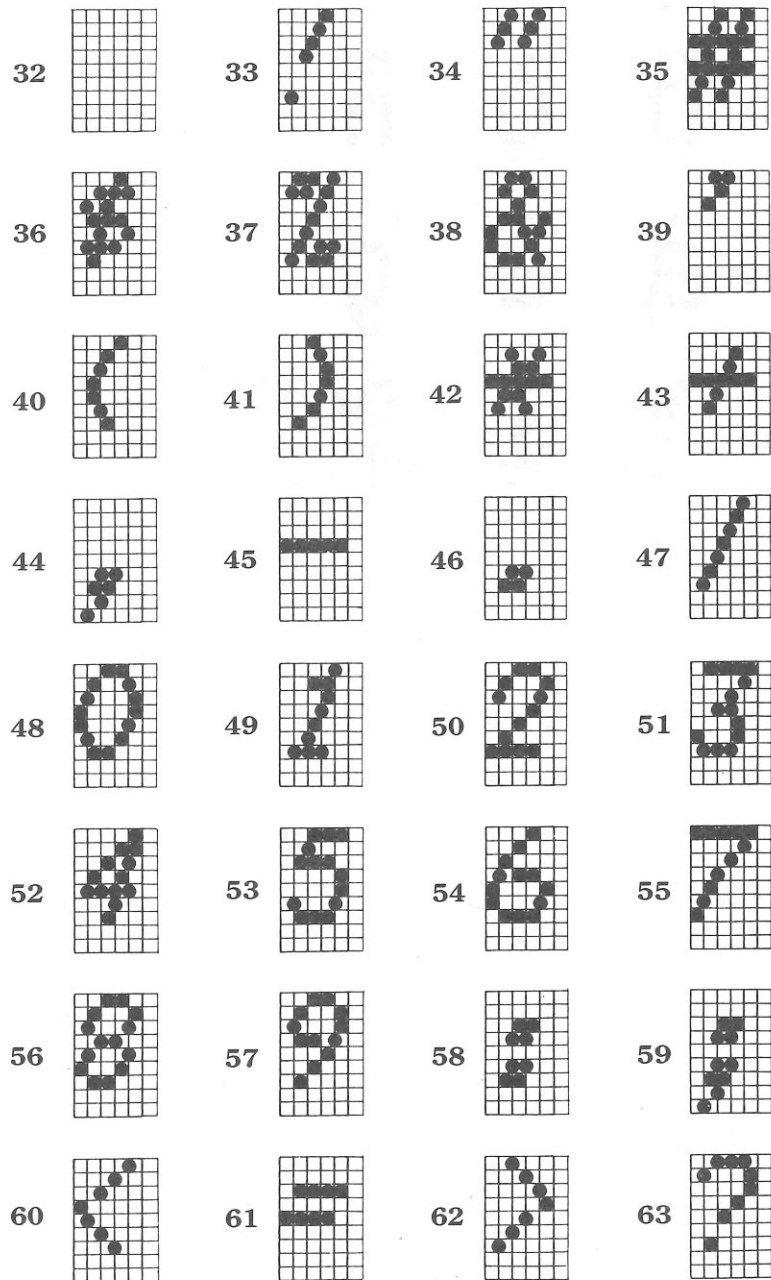
126

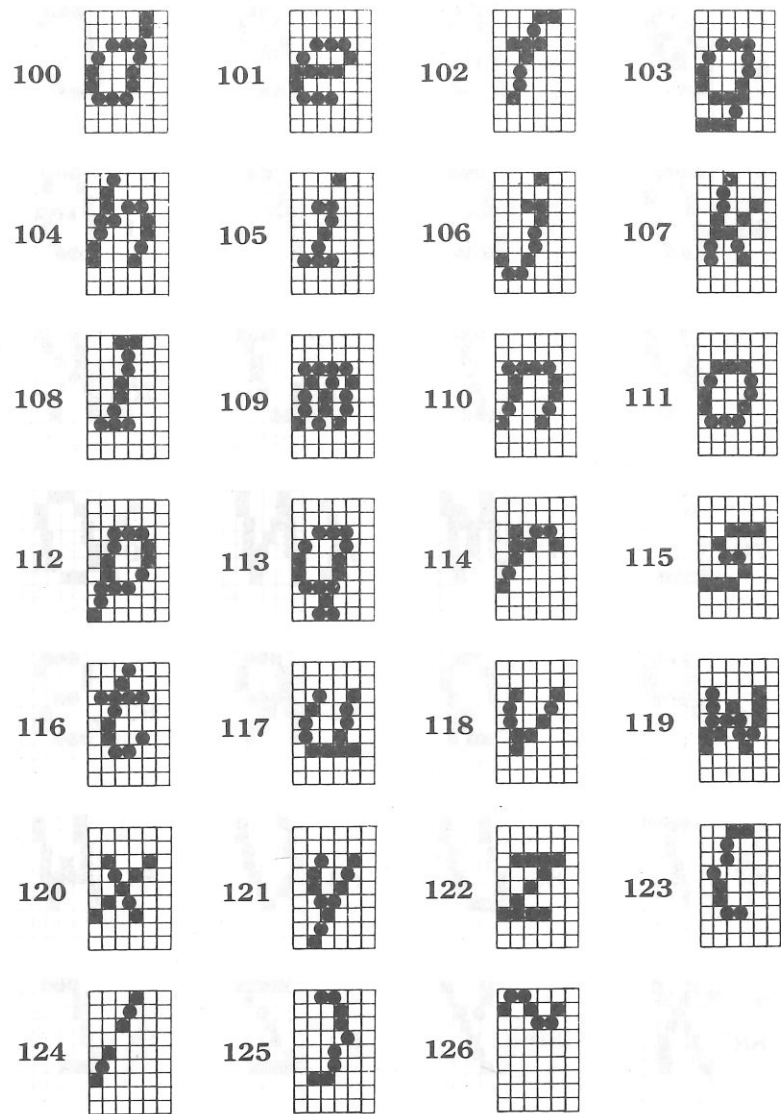




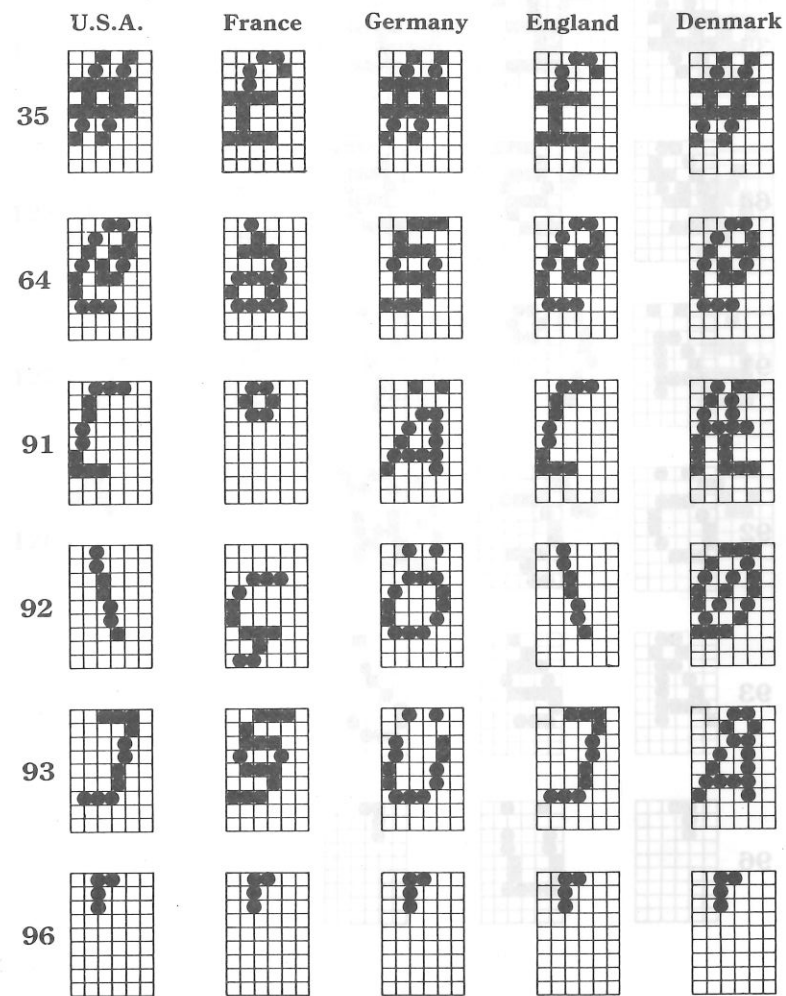
## ITALIC CHARACTERS

■ Standard characters





■ International characters

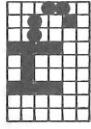


Sweden

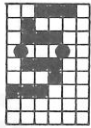
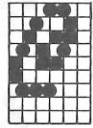
Italy

Spain

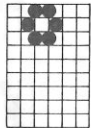
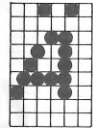
35



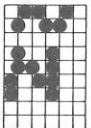
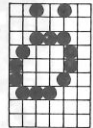
64



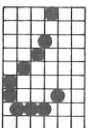
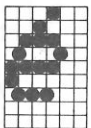
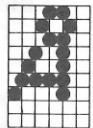
91



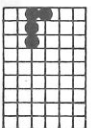
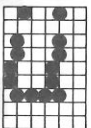
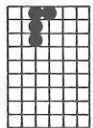
92



93



96



U.S.A.

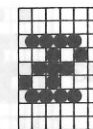
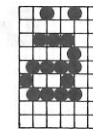
France

Germany

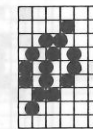
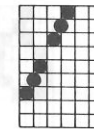
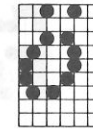
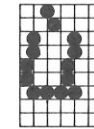
England

Denmark

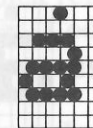
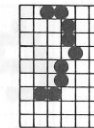
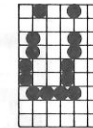
123



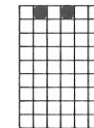
124

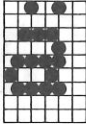

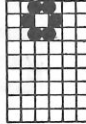
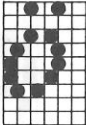

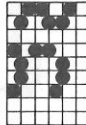
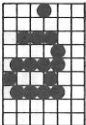


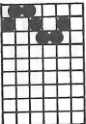
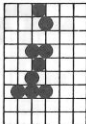
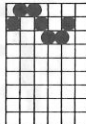


125



126



|     | Sweden                                                                            | Italy                                                                             | Spain                                                                             |
|-----|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| 123 |  |  |  |
| 124 |  |  |  |
| 125 |  |  |  |
| 126 |  |  |  |

## APPENDIX D

# 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 a "d", "n" or "m" to indicate a variable. You should insert the digit numbers or 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 italic characters.**

**CODE**                <ESC>    "4"  
(decimal ASCII)       27        52  
(hex ASCII)           1B        34

**REMARKS**           This command causes all characters to be printed in italics until italic printing is cancelled.

**SEE**                 Chapter 2

**PURPOSE**            **Cancels italic characters.**

**CODE**                <ESC>    "5"  
(decimal ASCII)       27        53  
(hex ASCII)           1B        35

**REMARKS**           This command causes the printer to cancel the italic printing and selects instead the standard roman characters.

**SEE**                 Chapter 2

**PURPOSE**           **Selects the American character set.**

**CODE**                <ESC>    "Z"       7        0  
(decimal ASCII)       27        90       7        0  
(hex ASCII)           1B        5A       07       00

**REMARKS**           This command selects the American character set.  
You can select the American character set as a power-on default by setting DIP switches 1-6, 1-7, and 1-8 on.

**SEE**                 Chapter 4

**PURPOSE**           **Selects the British character set.**

**CODE**                <ESC> "Z" 4 0 <ESC> "D" 3 0  
(decimal ASCII)       27 90 4 0 27 68 3 0  
(hex ASCII)           1B 5A 04 00 1B 44 03 00

**REMARKS**           This command selects the British character set.

You can select the British character set as a power-on default by setting DIP switches 1-6 and 1-7 off, and 1-8 on.

**SEE**                 Chapter 4

**PURPOSE**           **Selects the German character set.**

**CODE**                <ESC> "Z" 3 0 <ESC> "D" 4 0  
(decimal ASCII)       27 90 3 0 27 68 4 0  
(hex ASCII)           1B 5A 03 00 1B 44 04 00

**REMARKS**           This command selects the German character set.

You can select the German character set as a power-on default by setting DIP switches 1-6 and 1-8 on, and 1-7 off.

**SEE**                 Chapter 4

**PURPOSE**           **Selects the French character set.**

**CODE**                <ESC> "Z" 1 0 <ESC> "D" 6 0  
(decimal ASCII)       27 90 1 0 27 68 6 0  
(hex ASCII)           1B 5A 01 00 1B 44 06 00

**REMARKS**           This command selects the French character set.

You can select the French character set as a power-on default by setting DIP switches 1-7 and 1-8 on, and 1-6 off.

**SEE**                 Chapter 4

**PURPOSE**      **Selects the Swedish character set.**

|                 |           |    |    |           |    |    |
|-----------------|-----------|----|----|-----------|----|----|
| CODE            | <ESC> "Z" | 2  | 0  | <ESC> "D" | 5  | 0  |
| (decimal ASCII) | 27        | 90 | 2  | 0         | 27 | 68 |
| (hex ASCII)     | 1B        | 5A | 02 | 00        | 1B | 44 |

**REMARKS**      This command selects the Swedish character set.

You can select the Swedish character set as a power-on default by setting DIP switches 1-6 and 1-8 off, and 1-7 on.

**SEE**              Chapter 4

**PURPOSE**      **Selects the Italian character set.**

|                 |           |    |    |           |    |    |
|-----------------|-----------|----|----|-----------|----|----|
| CODE            | <ESC> "Z" | 6  | 0  | <ESC> "D" | 1  | 0  |
| (decimal ASCII) | 27        | 90 | 6  | 0         | 27 | 68 |
| (hex ASCII)     | 1B        | 5A | 06 | 00        | 1B | 44 |

**REMARKS**      This command selects the Italian character set.

You can select the Italian character set as a power-on default by setting DIP switches 1-7 and 1-8 off, and 1-6 on.

**SEE**              Chapter 4

**PURPOSE**      **Selects the Spanish character set.**

|                 |       |     |    |    |
|-----------------|-------|-----|----|----|
| CODE            | <ESC> | "D" | 7  | 0  |
| (decimal ASCII) | 27    | 68  | 7  | 0  |
| (hex ASCII)     | 1B    | 44  | 07 | 00 |

**REMARKS**      This command selects the Spanish character set.

You can select the Spanish character set as a power-on default by setting DIP switches 1-6, 1-7 and 1-8 off.

**SEE**              Chapter 4

**PURPOSE**      **Selects the Danish character set.**

|                 |           |    |    |           |    |    |
|-----------------|-----------|----|----|-----------|----|----|
| CODE            | <ESC> "Z" | 5  | 0  | <ESC> "D" | 2  | 0  |
| (decimal ASCII) | 27        | 90 | 5  | 0         | 27 | 68 |
| (hex ASCII)     | 1B        | 5A | 05 | 00        | 1B | 44 |

**REMARKS**      This command selects the Danish character set.

You can select the Danish character set as a power-on default by setting DIP switches 1-6 and 1-7 on, and 1-8 off.

**SEE**              Chapter 4

**PURPOSE**      **Selects NLQ characters.**

|                 |       |     |    |
|-----------------|-------|-----|----|
| CODE            | <ESC> | "a" | 2  |
| (decimal ASCII) | 27    | 97  | 2  |
| (hex ASCII)     | 1B    | 61  | 02 |

**REMARKS**      This command causes the printer to print near letter quality (NLQ) characters until NLQ mode is cancelled. NLQ mode cannot be used with any other special printing functions except underlining, expanded printing, and big character printing. This command is ignored when the "Panel" mode is selected at the power-on.

In addition, this command sets the print density to 80 dots/inch in the dot graphics mode.

**NOTE:** The character "2" (decimal code 50, hexadecimal code 32) can be used instead of ASCII 2.

**SEE**              Chapter 2

**PURPOSE** Cancels NLQ characters.

|                 |       |     |    |
|-----------------|-------|-----|----|
| <b>CODE</b>     | <ESC> | "a" | 0  |
| (decimal ASCII) | 27    | 97  | 0  |
| (hex ASCII)     | 1B    | 61  | 00 |

**REMARKS** This command cancels NLQ printing and returns the printer to the draft mode. 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 2

**PURPOSE** Cancels NLQ characters.

|                 |       |     |    |
|-----------------|-------|-----|----|
| <b>CODE</b>     | <ESC> | "a" | 1  |
| (decimal ASCII) | 27    | 97  | 1  |
| (hex ASCII)     | 1B    | 61  | 01 |

**REMARKS** Same as <ESC> "a" 0, above.

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

**SEE** Chapter 2

■ Font pitch controls

**PURPOSE** Sets the print pitch to pica.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "n" |
| (decimal ASCII) | 27    | 110 |
| (hex ASCII)     | 1B    | 6E  |

**REMARKS** This command causes printing to be done in pica pitch with 80 characters per line. This command is ignored when the "Panel" mode is selected at the power-on.

In addition, this command sets the print density to 72 dots/inch in the dot graphics mode.

**SEE** Chapter 2

**PURPOSE** Sets the print pitch to pica.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "N" |
| (decimal ASCII) | 27    | 78  |
| (hex ASCII)     | 1B    | 4E  |

**REMARKS** This command causes printing to be done in pica pitch with 80 characters per line. This command is ignored when the "Panel" mode is selected at the power-on.

In addition, this command sets the print density to 80 dots/inch in the dot graphics mode.

**SEE** Chapter 2

**PURPOSE** Sets the print pitch to elite.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "E" |
| (decimal ASCII) | 27    | 69  |
| (hex ASCII)     | 1B    | 45  |

**REMARKS** This command causes printing to be done in elite pitch with 96 characters per line (NLQ characters are not printed in elite pitch). This command is ignored when the "Panel" mode is selected at the power-on.

In addition, this command sets the print density to 90 dots/inch in the dot graphics mode.

**SEE** Chapter 2

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

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "e" |
| (decimal ASCII) | 27    | 101 |
| (hex ASCII)     | 1B    | 65  |

**REMARKS** This command causes printing to be done in condensed pitch with 136 characters per line (NLQ characters are not printed in condensed pitch). This command is ignored when the "Panel" mode is selected at the power-on. In addition, this command sets the print density to 120 dots/inch in the dot graphics mode.

**SEE** Chapter 2

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

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "q" |
| (decimal ASCII) | 27    | 113 |
| (hex ASCII)     | 1B    | 71  |

**REMARKS** Same as <ESC> "e", above.

**SEE** Chapter 2

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

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "Q" |
| (decimal ASCII) | 27    | 81  |
| (hex ASCII)     | 1B    | 51  |

**REMARKS** This command causes printing to be done in condensed pitch with 136 characters per line (NLQ characters are not printed in condensed pitch). This command is ignored when the "Panel" mode is selected at the power-on. In addition, this command sets the print density to 160 dots/inch in the dot graphics mode.

**SEE** Chapter 2

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

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "P" |
| (decimal ASCII) | 27    | 80  |
| (hex ASCII)     | 1B    | 50  |

**REMARKS** This command causes printing to be done in pica proportional pitch. This command is ignored when the "Panel" mode is selected at the power-on. In addition, this command sets the print density to 160 dots/inch in the dot graphics mode.

**SEE** Chapter 2



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

|                 |       |     |
|-----------------|-------|-----|
| CODE            | <ESC> | "p" |
| (decimal ASCII) | 27    | 112 |
| (hex ASCII)     | 1B    | 70  |

**REMARKS** This command causes printing to be done in elite proportional pitch. This command is ignored when the "Panel" mode is selected at the power-on. In addition, this command sets the print density to 160 dots/inch in the dot graphics mode.

**SEE** Chapter 2

**PURPOSE** Sets *n* dot spaces between proportional characters.

|                 |       |     |          |
|-----------------|-------|-----|----------|
| CODE            | <ESC> | "s" | <i>n</i> |
| (decimal ASCII) | 27    | 115 | <i>n</i> |
| (hex ASCII)     | 1B    | 73  | <i>n</i> |

**REMARKS** This command sets spacing between characters *n* dots in proportional characters. The value of *n* must be between 0 and 9.

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

|                 |      |
|-----------------|------|
| CODE            | <SO> |
| (decimal ASCII) | 14   |
| (hex ASCII)     | 0E   |

**REMARKS** This command causes characters to be printed twice as wide as normally (half the current pitch) until expanded printing is cancelled.

**SEE** Chapter 2

**PURPOSE** Cancels the expanded print.

|                 |      |
|-----------------|------|
| CODE            | <SI> |
| (decimal ASCII) | 15   |
| (hex ASCII)     | 0F   |

**REMARKS** This command resets the character pitch to what it was before expanded printing was set.

**SEE** Chapter 2

#### ■ Special print modes

**PURPOSE** Sets the master print mode.

|                 |       |     |          |
|-----------------|-------|-----|----------|
| CODE            | <ESC> | "i" | <i>n</i> |
| (decimal ASCII) | 27    | 105 | <i>n</i> |
| (hex ASCII)     | 1B    | 69  | <i>n</i> |

**REMARKS** This is a powerful command that allows the user to set several printing characteristics at one time: print pitch, condensed print, expanded print, emphasizing, boldface, underlining, and any combination of these as determined by *n*, a number from 0 to 255. (See Table 2-8 for details.)

**SEE** Chapter 2

**PURPOSE** Selects emphasized printing.

|                 |       |     |
|-----------------|-------|-----|
| CODE            | <ESC> | "!" |
| (decimal ASCII) | 27    | 33  |
| (hex ASCII)     | 1B    | 21  |

**REMARKS** This command causes characters to be emphasized until emphasized printing is cancelled.

**SEE** Chapter 2

**PURPOSE** Cancels emphasized printing.

|                 |       |    |
|-----------------|-------|----|
| <b>CODE</b>     | <ESC> | “” |
| (decimal ASCII) | 27    | 34 |
| (hex ASCII)     | 1B    | 22 |

**REMARKS** This command cancels emphasized printing.

**NOTE:** In the BASIC program, you must use CHR\$(34) instead of “”.

**SEE** Chapter 2

**PURPOSE** Selects boldface printing.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | “l” |
| (decimal ASCII) | 27    | 91  |
| (hex ASCII)     | 1B    | 5B  |

**REMARKS** This command causes characters to be printed in boldface until boldface is cancelled. Boldface cannot be used with superscripts or subscripts. This command is ignored when the “Panel” mode is selected at the power-on.

**SEE** Chapter 2

**PURPOSE** Cancels boldface printing.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | “j” |
| (decimal ASCII) | 27    | 93  |
| (hex ASCII)     | 1B    | 5D  |

**REMARKS** This command turns off boldface printing and returns the printer to normal printing. This command is ignored when the “Panel” mode is selected at the power-on.

**SEE** Chapter 2

**PURPOSE** Selects underlining.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | “X” |
| (decimal ASCII) | 27    | 88  |
| (hex ASCII)     | 1B    | 58  |

**REMARKS** This command underlines the following characters until underlining is cancelled.

**SEE** Chapter 2

**PURPOSE** Cancels underlining.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | “Y” |
| (decimal ASCII) | 27    | 89  |
| (hex ASCII)     | 1B    | 59  |

**REMARKS** This command stops underlining.

**SEE** Chapter 2

**PURPOSE** Selects superscripts.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | “x” |
| (decimal ASCII) | 27    | 120 |
| (hex ASCII)     | 1B    | 78  |

**REMARKS** This command raises the following characters and prints them as superscripts until superscripting is cancelled. Superscripts are printed from left to right only and in boldface. Superscripts cannot be used with NLQ printing.

**SEE** Chapter 2

**PURPOSE** Selects superscripts.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | “w” |
| (decimal ASCII) | 27    | 119 |
| (hex ASCII)     | 1B    | 77  |

**REMARKS** Same as <ESC> “x”, above.

**SEE** Chapter 2

|                 |                                                                                                                                                                                                         |     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>PURPOSE</b>  | <b>Selects subscripts.</b>                                                                                                                                                                              |     |
| <b>CODE</b>     | <ESC>                                                                                                                                                                                                   | "y" |
| (decimal ASCII) | 27                                                                                                                                                                                                      | 121 |
| (hex ASCII)     | 1B                                                                                                                                                                                                      | 79  |
| <b>REMARKS</b>  | This command lowers the following characters and prints as subscripts until subscripting is cancelled. All conditions described for superscripts also apply to subscripts.                              |     |
| <b>SEE</b>      | Chapter 2                                                                                                                                                                                               |     |
| <b>PURPOSE</b>  | <b> Cancels a superscript or subscript.</b>                                                                                                                                                             |     |
| <b>CODE</b>     | <ESC>                                                                                                                                                                                                   | "z" |
| (decimal ASCII) | 27                                                                                                                                                                                                      | 122 |
| (hex ASCII)     | 1B                                                                                                                                                                                                      | 7A  |
| <b>REMARKS</b>  | This command stops printing of superscripts or subscripts and sets normal printing. It also cancels uni-directional printing and boldface, which are set automatically for superscripts and subscripts. |     |
| <b>SEE</b>      | Chapter 2                                                                                                                                                                                               |     |
| <b>PURPOSE</b>  | <b> Cancels a superscript or subscript.</b>                                                                                                                                                             |     |
| <b>CODE</b>     | <ESC>                                                                                                                                                                                                   | "W" |
| (decimal ASCII) | 27                                                                                                                                                                                                      | 87  |
| (hex ASCII)     | 1B                                                                                                                                                                                                      | 57  |
| <b>REMARKS</b>  | Same as <ESC> "z", above.                                                                                                                                                                               |     |
| <b>SEE</b>      | Chapter 2                                                                                                                                                                                               |     |

## 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 and related commands

|                 |                                                                                                                                                                          |     |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>PURPOSE</b>  | <b>Sets the normal line feed direction.</b>                                                                                                                              |     |
| <b>CODE</b>     | <ESC>                                                                                                                                                                    | "f" |
| (decimal ASCII) | 27                                                                                                                                                                       | 102 |
| (hex ASCII)     | 1B                                                                                                                                                                       | 66  |
| <b>REMARKS</b>  | This command sets the line feed direction to the normal forward.                                                                                                         |     |
| <b>SEE</b>      | Chapter 3                                                                                                                                                                |     |
| <b>PURPOSE</b>  | <b>Sets the reverse line feed direction.</b>                                                                                                                             |     |
| <b>CODE</b>     | <ESC>                                                                                                                                                                    | "r" |
| (decimal ASCII) | 27                                                                                                                                                                       | 114 |
| (hex ASCII)     | 1B                                                                                                                                                                       | 72  |
| <b>REMARKS</b>  | This command sets the line feed direction to the reverse.                                                                                                                |     |
| <b>SEE</b>      | Chapter 3                                                                                                                                                                |     |
| <b>PURPOSE</b>  | <b>Advances or reverses the paper one line (line feed).</b>                                                                                                              |     |
| <b>CODE</b>     | <LF>                                                                                                                                                                     |     |
| (decimal ASCII) | 10                                                                                                                                                                       |     |
| (hex ASCII)     | 0A                                                                                                                                                                       |     |
| <b>REMARKS</b>  | This command feeds the paper depending on the previously set direction. The actual distance by the line feed is set through various codes which can be sent (see below). |     |
| <b>SEE</b>      | Chapter 3                                                                                                                                                                |     |

**PURPOSE** Sets line spacing to 1/6 inch.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "A" |
| (decimal ASCII) | 27    | 65  |
| (hex ASCII)     | 1B    | 41  |

**REMARKS** This command sets the actual distance the paper advances or reverses during all subsequent line feeds to 1/6 inch.

**SEE** Chapter 3

**PURPOSE** Sets line spacing to 1/8 inch.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "B" |
| (decimal ASCII) | 27    | 66  |
| (hex ASCII)     | 1B    | 42  |

**REMARKS** This command sets the actual distance the paper advances or reverses during all subsequent line feeds to 1/8 inch.

**SEE** Chapter 3

**PURPOSE** Sets line spacing to  $n/144$  inch.

|                 |       |     |       |
|-----------------|-------|-----|-------|
| <b>CODE</b>     | <ESC> | "T" | $d_2$ |
| (decimal ASCII) | 27    | 84  | $d_2$ |
| (hex ASCII)     | 1B    | 54  | $d_2$ |

**REMARKS** This command sets the actual distance the paper advances or reverses during all subsequent line feeds to  $n/144$  inch. The value of  $n$  is given with the two-digit numbers,  $d_2$ .

**SEE** Chapter 3

**PURPOSE** Sets line feed function without carriage return.

|                 |       |     |    |
|-----------------|-------|-----|----|
| <b>CODE</b>     | <ESC> | "1" | 1  |
| (decimal ASCII) | 27    | 108 | 1  |
| (hex ASCII)     | 1B    | 6C  | 01 |

**REMARKS** This command sets the line feed function without carriage return. When the <LF> command is sent to the printer after this command has been sent, the printer does not cause a carriage return function.

**SEE** Chapter 3

**PURPOSE** Sets the normal line feed function.

|                 |       |     |    |
|-----------------|-------|-----|----|
| <b>CODE</b>     | <ESC> | "1" | 0  |
| (decimal ASCII) | 27    | 108 | 0  |
| (hex ASCII)     | 1B    | 6C  | 00 |

**REMARKS** This command cancels the line feed function without carriage return set with the <ESC> "1" 1 command, and restores the normal line feed function.

**SEE** Chapter 3

**PURPOSE** No line feed at buffer full printing.

|                 |       |     |         |    |
|-----------------|-------|-----|---------|----|
| <b>CODE</b>     | <ESC> | "Z" | "space" | 0  |
| (decimal ASCII) | 27    | 90  | 32      | 0  |
| (hex ASCII)     | 1B    | 5A  | 20      | 00 |

**REMARKS** Normally, this printer causes a line feed when the buffer becomes full. If you want not to cause a line feed at buffer full printing, this command takes effect.

**PURPOSE** Adds line feed at buffer full printing.

|                 |       |     |         |    |
|-----------------|-------|-----|---------|----|
| CODE            | <ESC> | "D" | "space" | 0  |
| (decimal ASCII) | 27    | 68  | 32      | 0  |
| (hex ASCII)     | 1B    | 44  | 20      | 00 |

**REMARKS** This command cancels the <ESC> "Z" "space" 0 command, and restores to cause a line feed when the buffer becomes full.

■ 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 DIP switch 1-5 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 installed.

**SEE** Chapter 3

**PURPOSE** Sets page length to  $n/144$  inches.

|                 |       |     |       |
|-----------------|-------|-----|-------|
| CODE            | <ESC> | "H" | $d_4$ |
| (decimal ASCII) | 27    | 72  | $d_4$ |
| (hex ASCII)     | 1B    | 48  | $d_4$ |

**REMARKS** This command sets the length of all subsequent pages to  $n/144$  inches. The value of  $n$  is given the four-digit numbers,  $d_4$ . You can select a power-on default form length of 11 inches or 12 inches by setting DIP switch 1-5. This command is ignored when the optional automatic sheet feeder is installed.

**SEE** Chapter 3

**PURPOSE** Sets the top of form to the current position.

|                 |       |     |
|-----------------|-------|-----|
| CODE            | <ESC> | "v" |
| (decimal ASCII) | 27    | 118 |
| (hex ASCII)     | 1B    | 76  |

**REMARKS** This command sets the top of form to the current position.

**SEE** Chapter 3

**PURPOSE** Sets the vertical format unit.

|                 |      |       |      |
|-----------------|------|-------|------|
| CODE            | <GS> | ..... | <RS> |
| (decimal ASCII) | 29   | ..... | 30   |
| (hex ASCII)     | 1D   | ..... | 1E   |

**REMARKS** This command sets the top of form (TOF), vertical tab positions, bottom margins, and the form length. This command is ignored when the optional automatic sheet feeder is installed.

**SEE** Chapter 3

**PURPOSE** Selects vertical channel or feeds paper  $n$  lines.

|                 |      |     |
|-----------------|------|-----|
| CODE            | <US> | $d$ |
| (decimal ASCII) | 31   | $d$ |
| (hex ASCII)     | 1F   | $d$ |

**REMARKS** This command selects one of the multiple vertical channels determined by the value of  $d$  when  $d$  is between "A" and "F". When the value of  $d$  is between "1" and "?", this command causes the printer to feed paper  $n$  lines determined by the value of  $d$ .

**SEE** Chapter 3

## 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.

**SEE** Chapter 3

**PURPOSE** Sets the left margin.

**CODE** <ESC> "L"  $d_3$   
(decimal ASCII) 27 76  $d_3$   
(hex ASCII) 1B 4C  $d_3$

**REMARKS** This command sets the left margin to  $d_3$  characters. Each line will begin in the ( $d_3 + 1$ )th character position from the left edge. The value of  $d_3$  must be given with three-digit number. You can set the left margin manually with the control panel. **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 3

**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 not set at power-on.

**SEE** Chapter 3

**PURPOSE** Sets horizontal tab positions.

**CODE** <ESC> "("  $d1_3$  ","  $d2_3$  "," ...  $dx_3$  "."  
(decimal ASCII) 27 40  $d1_3$  44  $d2_3$  44 ...  $dx_3$  46  
(hex ASCII) 1B 28  $d1_3$  2C  $d2_3$  2C ...  $dx_3$  2E

**REMARKS** This command cancels all current horizontal tab positions and sets those defined at print positions  $d1_3$ ,  $d2_3$ ,  $d3_3$ , etc. The maximum number of horizontal tab positions allowed is 32. The number of horizontal tab positions should be given with three-digit number. The comma (,) is used to separate the tab positions and the period (.) is used to as a command terminator. Each horizontal tab position must be specified in ascending order.

**SEE** Chapter 3

|                                               |                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PURPOSE</b>                                | <b>Clears the selected tab positions.</b>                                                                                                                                                                                                                                                                                                                |
| <b>CODE</b><br>(decimal ASCII)<br>(hex ASCII) | <ESC> “y” $d1_3$ “,” $d2_3$ “,” ... $dx_3$ “.”<br>27 41 $d1_3$ 44 $d2_3$ 44 ... $dx_3$ 46<br>1B 29 $d1_3$ 2C $d2_3$ 2C ... $dx_3$ 2E                                                                                                                                                                                                                     |
| <b>REMARKS</b>                                | This command cancels the selected horizontal tab positions $d1_3$ , $d2_3$ , $d3_3$ , etc. The number of horizontal tab positions should be given with three-digit number. The comma (,) is used to separate the tab positions and the period (.) is used to as a command terminator. Each horizontal tab position must be specified in ascending order. |
| <b>SEE</b>                                    | Chapter 3                                                                                                                                                                                                                                                                                                                                                |
| <b>PURPOSE</b>                                | <b>Sets one additional tab position.</b>                                                                                                                                                                                                                                                                                                                 |
| <b>CODE</b><br>(decimal ASCII)<br>(hex ASCII) | <ESC> “u” $d_3$<br>27 117 $d_3$<br>1B 75 $d_3$                                                                                                                                                                                                                                                                                                           |
| <b>REMARKS</b>                                | This command sets horizontal tab in addition to those already set. This command can set only one additional tab, $d_3$ , at a time. The number of horizontal tab position should be given with three-digit number.                                                                                                                                       |
| <b>SEE</b>                                    | Chapter 3                                                                                                                                                                                                                                                                                                                                                |
| <b>PURPOSE</b>                                | <b>Clears all horizontal tab positions.</b>                                                                                                                                                                                                                                                                                                              |
| <b>CODE</b><br>(decimal ASCII)<br>(hex ASCII) | <ESC> “0”<br>27 48<br>1B 30                                                                                                                                                                                                                                                                                                                              |
| <b>REMARKS</b>                                | This command clears all horizontal tab positions already set.                                                                                                                                                                                                                                                                                            |
| <b>SEE</b>                                    | Chapter 3                                                                                                                                                                                                                                                                                                                                                |

|                                               |                                                                                                                                                                                                                         |
|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PURPOSE</b>                                | <b>Moves the print head to an absolute horizontal position.</b>                                                                                                                                                         |
| <b>CODE</b><br>(decimal ASCII)<br>(hex ASCII) | <ESC> “F” $d_4$<br>27 70 $d_4$<br>1B 46 $d_4$                                                                                                                                                                           |
| <b>REMARKS</b>                                | This command causes the printer to move the print head to an absolute horizontal position. The position, in dot unit, is determined by the value of $d_4$ . The number of $d_4$ should be given with four-digit number. |
| <b>PURPOSE</b>                                | <b>Moves the print head back one print position (backspace).</b>                                                                                                                                                        |
| <b>CODE</b><br>(decimal ASCII)<br>(hex ASCII) | <BS><br>8<br>08                                                                                                                                                                                                         |
| <b>REMARKS</b>                                | 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.                                          |
| <b>SEE</b>                                    | Chapter 4                                                                                                                                                                                                               |
| <b>PURPOSE</b>                                | <b>Sets alignment, or centering.</b>                                                                                                                                                                                    |
| <b>CODE</b><br>(decimal ASCII)<br>(hex ASCII) | <ESC> “J” $n$<br>27 74 $n$<br>1B 4A $n$                                                                                                                                                                                 |
| <b>REMARKS</b>                                | This command causes the printer to format text as follows:<br><br><u><math>n</math> Text formatting</u><br>0 Left-aligned (ragged right margin)<br>1 Centered<br>2 Right-aligned                                        |
| <b>SEE</b>                                    | Chapter 3                                                                                                                                                                                                               |

## DOWNLOAD CHARACTER COMMANDS

|                 |                                              |    |           |           |                                                                     |
|-----------------|----------------------------------------------|----|-----------|-----------|---------------------------------------------------------------------|
| PURPOSE         | <b>Defines download characters into RAM.</b> |    |           |           |                                                                     |
| CODE            | <ESC> "&"                                    | 0  | <i>n1</i> | <i>n2</i> | <i>m0</i> <i>m1</i> ... <i>m11</i><br>[ <i>m12</i> ... <i>m46</i> ] |
| (decimal ASCII) | 27 38                                        | 0  | <i>n1</i> | <i>n2</i> | <i>m0</i> <i>m1</i> ... <i>m11</i><br>[ <i>m12</i> ... <i>m46</i> ] |
| (hex ASCII)CODE | 1B 26                                        | 00 | <i>n1</i> | <i>n2</i> | <i>m0</i> <i>m1</i> ... <i>m11</i><br>[ <i>m12</i> ... <i>m46</i> ] |

**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 any number between 33 and 126. Following *n2* this printer expects character data bytes for each character to be defined. The first byte, *m0*, is the attribute bytes, for it specifies whether the character is a descender (if the first bit is 0), and the proportional width of the draft character (starting and ending dot columns are defined by the low order seven bits). *m1* through *m11* determine which dots form the draft character. In case of NLQ download characters, *m1* through *m46* determine which dots form the character.

**NOTE:** This command is ignored when the DIP switch 1-1 is set on.

**SEE** Chapter 5

|                 |                                                     |    |    |    |    |
|-----------------|-----------------------------------------------------|----|----|----|----|
| PURPOSE         | <b>Copies standard character ROM font into RAM.</b> |    |    |    |    |
| CODE            | <ESC> ":",                                          | 0  | 0  | 0  | 0  |
| (decimal ASCII) | 27 58                                               | 0  | 0  | 0  | 0  |
| (hex ASCII)     | 1B 3A                                               | 00 | 00 | 00 | 00 |

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

**NOTE:** This command is ignored when the DIP switch 1-1 is set on.

**SEE** Chapter 5

|                 |                                        |    |    |  |
|-----------------|----------------------------------------|----|----|--|
| PURPOSE         | <b>Selects download character set.</b> |    |    |  |
| 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 5

|                 |                                         |    |    |  |
|-----------------|-----------------------------------------|----|----|--|
| PURPOSE         | <b> Cancels download character set.</b> |    |    |  |
| 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 5



**DOT GRAPHICS COMMANDS**

|                 |                                   |     |       |                 |
|-----------------|-----------------------------------|-----|-------|-----------------|
| PURPOSE         | <b>Selects dot graphics mode.</b> |     |       |                 |
| CODE            | <ESC>                             | "G" | $d_4$ | $m1$ $m2$ ..... |
| (decimal ASCII) | 27                                | 71  | $d_4$ | $m1$ $m2$ ..... |
| (hex ASCII)     | 1B                                | 47  | $d_4$ | $m1$ $m2$ ..... |

REMARKS This command selects column-scan, bit-image graphics mode. The value of  $d_4$  represents the number of graphics characters to be printed with four-digit number. 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 6

PURPOSE **Selects dot graphics mode.**

|                 |       |     |       |                 |
|-----------------|-------|-----|-------|-----------------|
| CODE            | <ESC> | "S" | $d_4$ | $m1$ $m2$ ..... |
| (decimal ASCII) | 27    | 83  | $d_4$ | $m1$ $m2$ ..... |
| (hex ASCII)     | 1B    | 53  | $d_4$ | $m1$ $m2$ ..... |

REMARKS Same as <ESC> "G"  $d_4$ , above.

SEE Chapter 6

PURPOSE **Selects dot graphics mode.**

|                 |       |     |       |                 |
|-----------------|-------|-----|-------|-----------------|
| CODE            | <ESC> | "g" | $d_3$ | $m1$ $m2$ ..... |
| (decimal ASCII) | 27    | 103 | $d_3$ | $m1$ $m2$ ..... |
| (hex ASCII)     | 1B    | 67  | $d_3$ | $m1$ $m2$ ..... |

REMARKS This command selects column-scan, bit-image graphics mode. The value of  $d_3 \times 8$  represents the number of graphics characters to be printed with three-digit number. 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 6

PURPOSE **Repeats the dot pattern.**

|                 |       |     |       |     |
|-----------------|-------|-----|-------|-----|
| CODE            | <ESC> | "V" | $d_4$ | $n$ |
| (decimal ASCII) | 27    | 86  | $d_4$ | $n$ |
| (hex ASCII)     | 1B    | 56  | $d_4$ | $n$ |

REMARKS This command repeats a dot pattern determined by the value of  $n$  for  $d_4$  times. The number of  $d_4$  should be given with four-digit number.

**MACRO INSTRUCTION COMMANDS**

PURPOSE **Defines macro instruction.**

|                 |       |     |       |      |
|-----------------|-------|-----|-------|------|
| CODE            | <ESC> | "+" | ..... | <RS> |
| (decimal ASCII) | 27    | 43  | ..... | 30   |
| (hex ASCII)     | 1B    | 2B  | ..... | 1E   |

REMARKS This command cancels any existing macro instruction, and replace it with the instruction defined. The maximum number of characters allowed in the macro instruction is 16. The <RS> character marks the end of the macro definition.

SEE Chapter 4

PURPOSE **Executes macro instruction.**

|                 |       |     |    |
|-----------------|-------|-----|----|
| CODE            | <ESC> | "+" | 1  |
| (decimal ASCII) | 27    | 43  | 1  |
| (hex ASCII)     | 1B    | 2B  | 01 |

REMARKS This command executes a macro instruction that was previously defined.

SEE Chapter 4

**OTHER COMMANDS**

**PURPOSE** Sets the value of the eighth data bit to logical 0.

|                 |       |     |    |     |
|-----------------|-------|-----|----|-----|
| CODE            | <ESC> | "D" | 0  | "2" |
| (decimal ASCII) | 27    | 68  | 0  | 50  |
| (hex ASCII)     | 1B    | 43  | 00 | 32  |

**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.

**PURPOSE** Accepts the value of the eighth data bit as is.

|                 |       |     |    |     |
|-----------------|-------|-----|----|-----|
| CODE            | <ESC> | "Z" | 0  | "2" |
| (decimal ASCII) | 27    | 90  | 0  | 50  |
| (hex ASCII)     | 1B    | 5A  | 00 | 32  |

**REMARKS** This command cancels setting of the eighth data bit to logical 0. The printer will use the value of the eighth data bit that is sent from the computer.

**PURPOSE** Ends a line with <CR>, <LF>, <FF> or <HT>.

|                 |       |     |     |    |
|-----------------|-------|-----|-----|----|
| CODE            | <ESC> | "D" | "@" | 0  |
| (decimal ASCII) | 27    | 68  | 64  | 0  |
| (hex ASCII)     | 1B    | 43  | 40  | 00 |

**REMARKS** This command causes the printer to recognize <CR>, <LF>, <FF> and <HT> commands as end-of-line character.

**PURPOSE** Ends a line with <CR> only.

|                 |       |     |     |    |
|-----------------|-------|-----|-----|----|
| CODE            | <ESC> | "Z" | "@" | 0  |
| (decimal ASCII) | 27    | 90  | 64  | 0  |
| (hex ASCII)     | 1B    | 5A  | 40  | 00 |

**REMARKS** This command causes the printer to recognize only <CR> command as end-of-line character.

**PURPOSE****Prints "slash zero".**

|                 |       |     |    |    |
|-----------------|-------|-----|----|----|
| CODE            | <ESC> | "D" | 0  | 1  |
| (decimal ASCII) | 27    | 68  | 0  | 1  |
| (hex ASCII)     | 1B    | 44  | 00 | 01 |

**REMARKS** This command causes to print "zero" with slash.

**SEE** Chapter 4

**PURPOSE****Prints "normal zero".**

|                 |       |     |    |    |
|-----------------|-------|-----|----|----|
| CODE            | <ESC> | "Z" | 0  | 1  |
| (decimal ASCII) | 27    | 90  | 0  | 1  |
| (hex ASCII)     | 1B    | 5A  | 00 | 01 |

**REMARKS** This command cancels to print "slash zero" and returns to print "normal zero".

**SEE** Chapter 4

**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** Chapter 4

**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 4

**PURPOSE**                    **Disables paper-out detector.**

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "O" |
| (decimal ASCII) | 27    | 79  |
| (hex ASCII)     | 1B    | 4F  |

**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 1-2 can also set to disable the paper-out detector.

**SEE**                            Chapter 4

**PURPOSE**                    **Enables paper-out detector.**

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "o" |
| (decimal ASCII) | 27    | 111 |
| (hex ASCII)     | 1B    | 6F  |

**REMARKS**                    This command restores the function of the paper-out detector.

**SEE**                            Chapter 4

**PURPOSE**                    **Selects uni-directional printing.**

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | ">" |
| (decimal ASCII) | 27    | 62  |
| (hex ASCII)     | 1B    | 3E  |

**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.

**SEE**                            Chapter 4

**PURPOSE**                     **Cancels uni-directional printing.**

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "<" |
| (decimal ASCII) | 27    | 60  |
| (hex ASCII)     | 1B    | 3C  |

**REMARKS**                    This command cancels uni-directional printing and returns to the standard bi-directional printing, which is considerably faster.

**SEE**                            Chapter 4

**PURPOSE**                    **Enlarges characters in whole or in part; cancels same.**

|                 |       |     |          |
|-----------------|-------|-----|----------|
| <b>CODE</b>     | <ESC> | "h" | <i>n</i> |
| (decimal ASCII) | 27    | 104 | <i>n</i> |
| (hex ASCII)     | 1B    | 68  | <i>n</i> |

**REMARKS**                    This special command enlarges characters following the command until the enlargement is cancelled. The values of *n* have the following effects.

| <i>n</i> | Effect                         |
|----------|--------------------------------|
| 0        | Cancels enlargement            |
| 1        | Double-high, double-wide       |
| 2        | Quadruple-high, quadruple-wide |

**SEE**                            Chapter 4

**PURPOSE** Resets the printer.

|                 |       |     |
|-----------------|-------|-----|
| <b>CODE</b>     | <ESC> | "c" |
| (decimal ASCII) | 27    | 99  |
| (hex ASCII)     | 1B    | 63  |

**REMARKS** This command reinitializes the printer. The print buffer is cleared, and the form length, bottom margin, and international character set are all reset to the values defined by their respective DIP switches. The main difference between the <ESC> "c" command and turning the printer off and back on again is that download characters and macro instructions are preserved with this command.

**SEE** Chapter 4

**PURPOSE** Selects auto feed mode.

|                 |       |      |    |
|-----------------|-------|------|----|
| <b>CODE</b>     | <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.

**SEE** Chapter 4

**PURPOSE** Selects auto feed mode.

|                 |     |     |     |     |     |
|-----------------|-----|-----|-----|-----|-----|
| <b>CODE</b>     | "(" | "(" | "4" | "") | "") |
| (decimal ASCII) | 40  | 40  | 52  | 41  | 41  |
| (hex ASCII)     | 28  | 28  | 34  | 29  | 29  |

**REMARKS** The same as <ESC> <EM> 4, above.

**SEE** Chapter 4

**PURPOSE** Cancels auto feed mode.

|                 |       |      |    |
|-----------------|-------|------|----|
| <b>CODE</b>     | <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.

**SEE** Chapter 4

**PURPOSE** Cancels auto feed mode.

|                 |     |     |     |     |     |
|-----------------|-----|-----|-----|-----|-----|
| <b>CODE</b>     | "(" | "(" | "0" | "") | "") |
| (decimal ASCII) | 40  | 40  | 48  | 41  | 41  |
| (hex ASCII)     | 28  | 28  | 30  | 29  | 29  |

**REMARKS** The same as <ESC> <EM> 0, above.

**SEE** Chapter 4

**PURPOSE** Supplies paper.

|                 |       |      |    |
|-----------------|-------|------|----|
| <b>CODE</b>     | <ESC> | <EM> | 1  |
| (decimal ASCII) | 27    | 25   | 1  |
| (hex ASCII)     | 1B    | 19   | 01 |

**REMARKS** This command causes the printer to supply paper under non-auto sheet feeding mode. This command is ignored when the optional automatic sheet feeder is not mounted.

**SEE** Chapter 4

**PURPOSE** Supplies paper.

|                 |     |     |     |     |     |
|-----------------|-----|-----|-----|-----|-----|
| <b>CODE</b>     | "(" | "(" | "1" | "") | "") |
| (decimal ASCII) | 40  | 40  | 49  | 41  | 41  |
| (hex ASCII)     | 28  | 28  | 31  | 29  | 29  |

**REMARKS** The same as <ESC> <EM> 1, above.

**SEE** Chapter 4

|                 |                                                                                                                                  |                   |            |            |            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------|-------------------|------------|------------|------------|
| <b>PURPOSE</b>  | <b>Ejects paper.</b>                                                                                                             |                   |            |            |            |
| <b>CODE</b>     | <b>&lt;ESC&gt;</b>                                                                                                               | <b>&lt;EM&gt;</b> | <b>"R"</b> |            |            |
| (decimal ASCII) | 27                                                                                                                               | 25                | 82         |            |            |
| (hex ASCII)     | 1B                                                                                                                               | 19                | 52         |            |            |
| <b>REMARKS</b>  | This command causes the printer to eject paper. This command is ignored when the optional automatic sheet feeder is not mounted. |                   |            |            |            |
| <b>SEE</b>      | Chapter 4                                                                                                                        |                   |            |            |            |
| <b>PURPOSE</b>  | <b>Ejects paper.</b>                                                                                                             |                   |            |            |            |
| <b>CODE</b>     | <b>"("</b>                                                                                                                       | <b>"("</b>        | <b>"R"</b> | <b>"("</b> | <b>"("</b> |
| (decimal ASCII) | 40                                                                                                                               | 40                | 82         | 41         | 41         |
| (hex ASCII)     | 28                                                                                                                               | 28                | 52         | 29         | 29         |
| <b>REMARKS</b>  | The same as <ESC> <EM> "R", above.                                                                                               |                   |            |            |            |
| <b>SEE</b>      | Chapter 4                                                                                                                        |                   |            |            |            |

---



---

## APPENDIX E

# COMMAND SUMMARY

## IN NUMERIC ORDER

---



---

| <b>Control code</b>                                       | <b>Function</b>                                             |
|-----------------------------------------------------------|-------------------------------------------------------------|
| 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)                                                 | Advances or reverses the paper one line (line feed).        |
| 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.                         |
| CHR\$(15)                                                 | Cancels the expanded print.                                 |
| CHR\$(24)                                                 | Cancels line.                                               |
| CHR\$(27)                                                 | Escape (indicated as <ESC> below.)                          |
| CHR\$(29) ..... CHR\$(30)                                 | Sets the vertical format unit.                              |
| CHR\$(31) <i>d</i>                                        | Selects vertical channel or feeds paper <i>n</i> lines.     |
| <ESC> CHR\$(25) 0                                         | Cancels auto feed mode.                                     |
| <ESC> CHR\$(25) 1                                         | Supplies paper.                                             |
| <ESC> CHR\$(25) 4                                         | Selects auto feed mode.                                     |
| <ESC> CHR\$(25) "R"                                       | Ejects paper.                                               |
| <ESC> "!"                                                 | Selects emphasized printing.                                |
| <ESC> CHR\$(34)                                           | Cancels emphasized printing.                                |
| <ESC> "%" 0 0                                             | Cancels download character set.                             |
| <ESC> "%" 1 0                                             | Selects download character set.                             |
| <ESC> "&" 0 <i>n1 n2 m0 m1 ... m46</i>                    | Defines download characters into RAM.                       |
| <ESC> "(" <i>d1</i> " ," <i>d2</i> " ," ... <i>dx</i> " " | Sets horizontal tab positions.                              |

|                                             |                                                          |
|---------------------------------------------|----------------------------------------------------------|
| ⟨ESC⟩ “)” $d_1$ “,” $d_2$ “,” ... $d_x$ “.” | Clears the selected tab positions.                       |
| ⟨ESC⟩ “+” 1                                 | Executes macro instruction.                              |
| ⟨ESC⟩ “+” ..... ⟨RS⟩                        | Defines macro instruction.                               |
| ⟨ESC⟩ “0”                                   | Clears all horizontal tab positions.                     |
| ⟨ESC⟩ “4”                                   | Selects italic characters.                               |
| ⟨ESC⟩ “5”                                   | Cancelles italic characters.                             |
| ⟨ESC⟩ “.” 0 0 0                             | Copies standard character ROM font into RAM.             |
| ⟨ESC⟩ “)”                                   | Selects uni-directional printing.                        |
| ⟨ESC⟩ “⟨”                                   | Cancelles uni-directional printing.                      |
| ⟨ESC⟩ “A”                                   | Sets line spacing to 1/6 inch.                           |
| ⟨ESC⟩ “B”                                   | Sets line spacing to 1/8 inch.                           |
| ⟨ESC⟩ “D” 0 1                               | Prints “slash zero”.                                     |
| ⟨ESC⟩ “D” 0 “2”                             | Sets the value of the eighth data bit to logical 0.      |
| ⟨ESC⟩ “D” 7 0                               | Selects the Spanish character set.                       |
| ⟨ESC⟩ “D” CHR\$(32) 0                       | Adds line feed at buffer full printing.                  |
| ⟨ESC⟩ “D” “@” 0                             | Ends a line with ⟨CR⟩, ⟨LF⟩, ⟨FF⟩, or ⟨HT⟩.              |
| ⟨ESC⟩ “E”                                   | Sets the print pitch to elite.                           |
| ⟨ESC⟩ “F” $d_4$                             | Moves the print head to an absolute horizontal position. |
| ⟨ESC⟩ “G” $d_4$ $m_1$ $m_2$ .....           | Selects dot graphics mode.                               |
| ⟨ESC⟩ “H” $d_4$                             | Sets page length to $n/144$ inches.                      |
| ⟨ESC⟩ “J” $n$                               | Sets alignment, or centering.                            |
| ⟨ESC⟩ “L” $d_3$                             | Sets the left margin.                                    |
| ⟨ESC⟩ “N”                                   | Sets the print pitch to pica.                            |
| ⟨ESC⟩ “O”                                   | Disables paper-out detector.                             |
| ⟨ESC⟩ “P”                                   | Sets the printer to pica proportional print.             |
| ⟨ESC⟩ “Q”                                   | Sets the printer to condensed print.                     |
| ⟨ESC⟩ “S” $d_4$ $m_1$ $m_2$ .....           | Selects dot graphics mode.                               |
| ⟨ESC⟩ “T” $d_2$                             | Sets line spacing to $n/144$ inch.                       |
| ⟨ESC⟩ “V” $d_4$ $n$                         | Repeats the dot pattern.                                 |
| ⟨ESC⟩ “W”                                   | Cancelles a superscript or subscript.                    |
| ⟨ESC⟩ “X”                                   | Selects underlining.                                     |
| ⟨ESC⟩ “Y”                                   | Cancelles underlining.                                   |
| ⟨ESC⟩ “Z” 0 1                               | Prints “normal zero”.                                    |
| ⟨ESC⟩ “Z” 0 “2”                             | Accepts the value of the eighth data bit as is.          |

|                                   |                                                          |
|-----------------------------------|----------------------------------------------------------|
| ⟨ESC⟩ “Z” 1 0 ⟨ESC⟩ “D” 6 0       | Selects the French character set.                        |
| ⟨ESC⟩ “Z” 2 0 ⟨ESC⟩ “D” 5 0       | Selects the Swedish character set.                       |
| ⟨ESC⟩ “Z” 3 0 ⟨ESC⟩ “D” 4 0       | Selects the German character set.                        |
| ⟨ESC⟩ “Z” 4 0 ⟨ESC⟩ “D” 3 0       | Selects the British character set.                       |
| ⟨ESC⟩ “Z” 5 0 ⟨ESC⟩ “D” 2 0       | Selects the Danish character set.                        |
| ⟨ESC⟩ “Z” 6 0 ⟨ESC⟩ “D” 1 0       | Selects the Italian character set.                       |
| ⟨ESC⟩ “Z” 7 0                     | Selects the American character set.                      |
| ⟨ESC⟩ “Z” CHR\$(32) 0             | No line feed at buffer full printing.                    |
| ⟨ESC⟩ “Z” “@” 0                   | Ends a line with ⟨CR⟩ only.                              |
| ⟨ESC⟩ “[”                         | Selects boldface printing.                               |
| ⟨ESC⟩ “]”                         | Cancelles boldface printing.                             |
| ⟨ESC⟩ “a” 0                       | Cancelles NLQ characters.                                |
| ⟨ESC⟩ “a” 1                       | Cancelles NLQ characters.                                |
| ⟨ESC⟩ “a” 2                       | Selects NLQ characters.                                  |
| ⟨ESC⟩ “c”                         | Resets the printer.                                      |
| ⟨ESC⟩ “e”                         | Sets the printer to condensed print.                     |
| ⟨ESC⟩ “f”                         | Sets the normal line feed direction.                     |
| ⟨ESC⟩ “g” $d_3$ $m_1$ $m_2$ ..... | Selects dot graphics mode.                               |
| ⟨ESC⟩ “h” $n$                     | Enlarges characters in whole or in part; cancelles same. |
| ⟨ESC⟩ “i” $n$                     | Sets the master print mode.                              |
| ⟨ESC⟩ “I” 0                       | Sets the normal line feed function.                      |
| ⟨ESC⟩ “I” 1                       | Sets line feed function without carriage return.         |
| ⟨ESC⟩ “n”                         | Sets the print pitch to pica.                            |
| ⟨ESC⟩ “o”                         | Enables paper-out detector.                              |
| ⟨ESC⟩ “p”                         | Sets the printer to elite proportional print.            |
| ⟨ESC⟩ “q”                         | Sets the printer to condensed print.                     |
| ⟨ESC⟩ “r”                         | Sets the reverse line feed direction.                    |
| ⟨ESC⟩ “s” $n$                     | Sets $n$ dot spaces between proportional characters.     |
| ⟨ESC⟩ “u” $d_3$                   | Sets one additional tab position.                        |
| ⟨ESC⟩ “v”                         | Sets the top of form to the current position.            |
| ⟨ESC⟩ “w”                         | Selects superscripts.                                    |

|           |                                    |
|-----------|------------------------------------|
| <ESC> "x" | Selects superscripts.              |
| <ESC> "y" | Selects subscripts.                |
| <ESC> "z" | Cancel a superscript or subscript. |
| "((0))"   | Cancel auto feed mode.             |
| "((1))"   | Supplies paper.                    |
| "((4))"   | Selects auto feed mode.            |
| "((R))"   | Ejects paper.                      |

---



---

## APPENDIX F

# TECHNICAL SPECIFICATIONS

---



---

|                    |                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Printing direction | Bi-directional, logic seeking<br>Uni-directional in dot graphics                                                                                                                                                                                                                                                                                                     |
| Character sets     | 96 standard ASCII characters<br>28 standard international characters<br>[8 sets]<br>96 near letter quality (NLQ) characters<br>28 NLQ international characters<br>[8 sets]<br>96 italic characters<br>28 italic international characters<br>[8 sets]<br>96 NLQ italic characters<br>28 NLQ italic international characters<br>[8 sets]<br>96 user-defined characters |
| Character matrix   | 18 × 23 dots, NLQ characters<br>9 × 11 dots, draft characters<br>8 × 576 dots, 72 DPI (dot per inch)<br>graphics<br>8 × 640 dots, 80 DPI graphics<br>8 × 720 dots, 90 DPI graphics<br>8 × 960 dots, 120 DPI graphics<br>8 × 1280 dots, 160 DPI graphics                                                                                                              |
| Line spacing       | 1/6 inch standard<br>1/8, or <i>n</i> /144 inch programmable                                                                                                                                                                                                                                                                                                         |
| Column width       | 80, normal pica<br>96, normal elite<br>136, normal condensed<br>40, expanded pica<br>48, expanded elite<br>68, expanded condensed<br>and proportional spacing                                                                                                                                                                                                        |

**Interface  
Special features**

Apple-serial interface  
Near Letter Quality  
Short tear-off  
Easy access format switches  
Self-test  
Hex dump  
Skip over perforation  
Automatic sheet feeder (option)

---

---

## APPENDIX G

# SERIAL INTERFACE SPECIFICATIONS

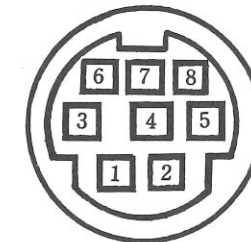
---

---

This interface cartridge provides the serial interface used to communicate with the Apple computers. The operating specifications of the interface are as follows:

|                     |                                                                           |
|---------------------|---------------------------------------------------------------------------|
| Data transfer rate: | 9600 baud (bits per second)                                               |
| Word length:        | 1 start bit<br>7 or 8 data bits<br>No parity<br>1 stop bit                |
| Signal levels:      | Mark or logical 1, -3 to -15 volts<br>Space or logical 0, +3 to +15 volts |
| Handshaking:        | DTR protocol mode                                                         |

The interface cartridge has a 8-pin mini-circular connector to connect to a computer. The functions of the pins are summarized in Table G-1.



**Figure G-1.** Front view of interface connector.



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

| Pin No. | Signal Name | Direction | Function                                                         |
|---------|-------------|-----------|------------------------------------------------------------------|
| 1       | DTR         | OUT       | The printer turns this pin ON when it is ready to receive data.  |
| 2       | N/C         |           | Unused.                                                          |
| 3       | N/C         |           | Unused.                                                          |
| 4       | GND         | —         | Printer's chassis ground.                                        |
| 5       | RXD         | IN        | This pin carries data to the printer.                            |
| 6       | N/C         |           | Unused.                                                          |
| 7       | N/C         |           | Unused.                                                          |
| 8       | RXDR        | IN        | This pin is the return path for data transmitted to the printer. |

**MEMO**

## Consumer Response

Star Micronics Co., Ltd. invites your suggestions and comments on this publication and the printer that it was written for. Please address your correspondence to:

*For Worldwide Headquarters:*

STAR MICRONICS CO., LTD.  
194 Nakayoshida  
Shizuoka, JAPAN 422-91  
Attn: Product Manager

*For American Market:*

STAR MICRONICS AMERICA INC.  
P.O. Box 1630  
El Toro, California, U.S.A. 92630  
Attn: Product Manager

*For European Market:*

STAR MICRONICS DEUTSCHLAND GMBH  
Frankfurter Allee 1-3.  
D-6236 Eschborn/Ts., W. Germany  
Attn: Product Manager

*For U.K Market:*

STAR MICRONICS U.K. LTD.  
Craven House, 4th. Floor  
40 Uxbridge Road, Ealing W.5, London  
Attn: Product Manager

*For Asian Market:*

STAR MICRONICS (S.E.A.) LTD.  
Rm 2407-8 Sincere Building;  
173 Des Voeux Road Central, Hong Kong  
Attn: Product Manager



**THE POWER BEHIND THE PRINTED WORD.**

PRINTED IN JAPAN

