

NB-15

USER'S MANUAL

NOT INTENDED FOR SALE

PN 80821206

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

NB-15, grafstar: Star Micronics Co., Ltd.

Apple, Apple II, Apple II+, Apple IIe, Applesoft: Apple computer Inc.

Commodore C-64: Commodore Business Machines, Inc.

Compaq: Compaq Computer Corporation

CP/M: Digital Research

EasyWriter: Information Unlimited Software, Inc.

IBM Personal Computer, IBM PC: International Business Machines Corp.

Kaypro: Kaypro Corporation

Microsoft BASIC: Microsoft Corporation

Osborne 1: Osborne Computer Corporation

WordStar: MicroPro International Corporation

TRS-80: Radio Shack, a division of Tandy Corporation

SuperCalc: Sorcim Corporation

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.

A Special Message to the New Owner

Congratulations for choosing this printer, one of the finest available at any price. Your new printer will reward you with many years of high-quality document printing, and is backed by Star Micronics' more than 30 years in manufacturing precision parts. You already know that it represents the state of the art in tack-sharp dot matrix printing with its rugged 24-pin print head. Its bidirectional, logic-seeking operation gives you Letter-Quality hardcopy at 100 characters per second, and draft-quality print at speeds of up to 300 characters per second.

Its many features include:

- High-resolution graphics
- Built-in expandable buffer
- An entire range of character sets for printing in eleven languages
- Choice of various typestyles
- Easy, front panel operation
- Automatic paper feed

And, best of all, your new printer is compatible with the most popular personal computers and software packages—so no matter how sophisticated your applications become, it will continue to meet your needs.

These are just a few of the reasons you chose a product by Star Micronics, one of the most respected names in dot matrix printers. We want you to get the most from your printer with the least effort on your part, so please take a few minutes to read this easy-to-understand guide. You'll find that it will help you produce crisp, clean documents in the shortest time possible.

You will also find that this printer is remarkably simple to set up and operate, and that it's virtually maintenance-free. Enjoy your new Star Micronics printer!

Table of Contents

Chapter 1	Setting Up Your Printer	1
	Where shall we put it?	
	What have we here?	
	Removing the printer cover	
	Removing the packing tube	
	Removing shipping screws	
	Installing the platen knob	
	Installing the ribbon cartridge	
	Installing the mute cover	
	Installing the interface board	
	Installing the font cartridge	
Chapter 2	Getting to Know Your Printer	9
	Controls and parts of the printer	
	Parts of the printer	
	Controls and indicators	
	Extra functions	
	Other controls	
	Selecting and loading paper	
	Loading single sheets	
	Loading sprocket-feed paper	
	Adjusting the print head	
	Extra functions with the control panel	
	Self-tests	
	Hex dump	
	Panel modes	
	Setting print start position	
	Setting the top of form	
	Setting the left and right margins	
	Connecting the printer	
Chapter 3	Using Commercial Software	27
	First, some terminology	
	Using your printer with SuperCalc	
	Using your printer with word processors	

Chapter 4	Controlling with BASIC Some basics about BASIC First steps Establishing communications The CHR\$ function Control codes The escape codes A note on command syntax Some problem codes	33
Chapter 5	Printing Text Some special kinds of text Letter Quality characters Characters in the Font Cartridge Underlines and overlines Boldface printing Superscripts and subscripts Changing the print pitch Expanded print Condensed print Proportional printing Mixing print modes	39
Chapter 6	Formatting Text Lines and line spacing Starting a new line Changing the line spacing Moving down the page without a carriage return Page control Form feed Changing the page length Bottom margin Setting left and right margins Horizontal and vertical tabs Horizontal tabs One-time horizontal tabs Vertical tabs Vertical tab channels	53

Chapter 7	Special Features of the Printer	73
	Now hear this	
	Resetting the printer	
	Putting your printer to sleep	
	Printing the bottom of the sheet	
	Backspace, delete, and cancel text	
	Immediate-print	
	Adjusting the width of space between characters	
	Repeat characters	
	Uni-directional printing	
	The seven bit dilemma	
	Block graphics characters and special symbols	
	International character sets	
	Printing BIG characters	
	The optional sheet feeder	
	Reading a hex dump	

Chapter 8	Creating Your Own Characters	89
	Dot matrix printing	
	The print matrix	
	Defining your own characters	
	Rule 1: Dots cannot overlap	
	Rule 2: Each row is divided into three bytes	
	Assigning a value of character space	
	Assigning a value to your character	
	Download character definition command	
	Printing download characters	
	How to print the characters in control code area	
	The print mode and the download characters	
	Draft download characters	
	Letter quality download characters	

Chapter 9	Dot Graphics	107
	Comparing dot graphics with download characters	
	Using the dot graphics command	
	Specifying the number of columns of dots	
	Specifying the graphics data	
	Combining text and graphics	
	Printing a design or logo	
	Plotting with your printer	
	How the program works	
	Using the 24-pin graphics mode	
	Compatibility with existing software	
	Redefining alternate graphics codes	
Chapter 10	Caring for Your Printer	127
	Cleaning the printer	
	Replacing the ribbon	
	Replacing the print head	
Appendix A	DIP Switch Settings	133
	Switch functions	
Appendix B	ASCII Codes and Conversion Chart	139
Appendix C	Function Codes	147
	Commands to control print style	
	Font style controls	
	Font pitch controls	
	Special print modes	
	Controlling the vertical print position	
	Line feed controls	
	Form feed and related commands	
	Bottom margins and vertical tabs	
	Controlling the horizontal print position	
	Download character commands	
	Dot graphics commands	
	Other commands	
Appendix D	Command Summary in Numeric Order	191
Appendix E	Technical Specifications	195
Appendix F	The Parallel Interface	199
	Functions of the Connector Signals	

Appendix G	Serial Interface Specifications	203
	Configuring the serial interface	
	The serial protocols	
	Serial busy protocols	
	XON/XOFF protocol	
	ACK protocol	
Appendix H	Connecting with Computer	209
	Connecting with IBM-PC and Compaq	
	BASIC programming	
	Listing programs	
	Connecting with Apple II computers	
	Applesoft BASIC	
	Listing programs	
	Connecting with TRS-80 computers	
	TRS-80 BASIC	
	Listing programs	
	Connecting with Kaypro, Osborne, and other	
	CP/M computers	
	Using MBASIC	
	Listing programs	

CHAPTER 1

SETTING UP YOUR PRINTER

Subjects we'll cover in Chapter 1 include—

- Choosing a suitable place for your printer;
- Unpacking your new printer;
- Setting it up.

WHERE SHALL WE PUT IT?

Before you do anything else, give some thought to where you'll be using your printer. Obviously, it will be somewhere near your computer. And both printer and computer will lead longer, healthier lives if they like their surroundings. For instance, we recommend...

- Using the printer on a flat surface.
- Keeping it out of direct sunlight and away from heat-producing appliance.
- Using it only in temperatures where you are comfortable.
- Avoiding locations with a lot of dust, grease, or humidity.
- Giving it "clean" electricity. Don't connect it to the same circuit used by large, noise-producing appliances (such as refrigerators).
- Making sure the line voltage is within 10% of the voltage specified on the identification plate.

WHAT HAVE WE HERE?

Let's open the carton and take a look at what's inside. Check each item in the box against Figure 1-1 to make sure that you have everything (there should be six items).

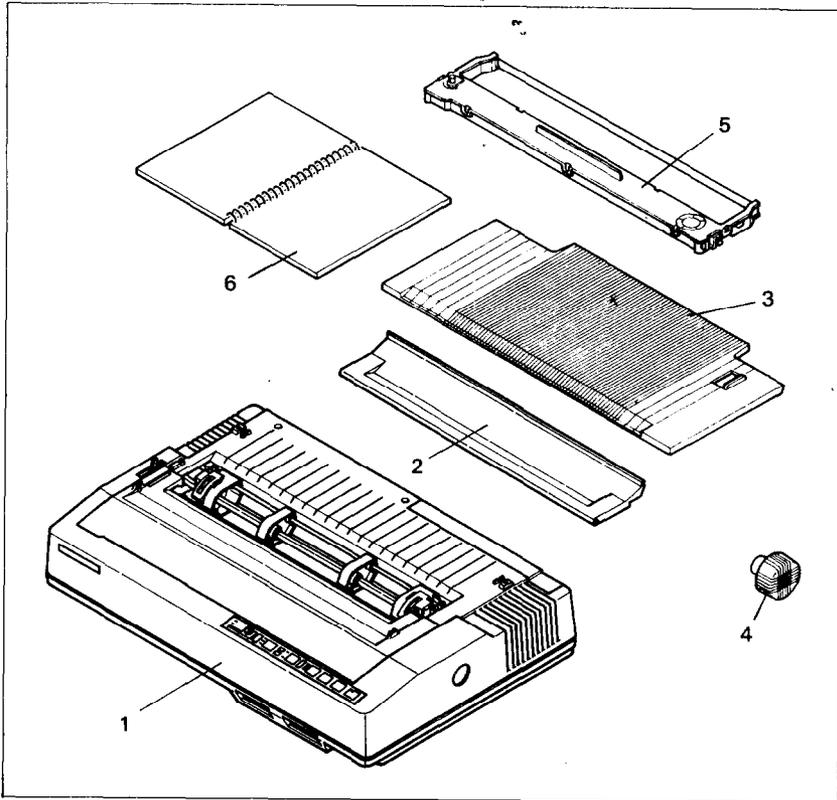


Figure 1-1. Check to make sure you have all six items: 1) Printer, 2) Mute cover, 3) Paper guide, 4) Platen knob, 5) Ribbon cartridge, and 6) User's manual.

You should also have a parallel interface board to connect your computer to the printer. Also available are a parallel interface board with extra buffer memory and a serial interface board, both optional. (More on the interfaces later.)

■ Removing the printer cover

The printer's cover is important for two reasons — it keeps dust and dirt away from the printer's delicate mechanism, and it absorbs nearly all of the printer's operating sounds. Don't take off the cover unless you have to change the ribbon or make an adjustment.

Removing the printer cover is easy. Lift up the back of the cover to disengage the three tabs at the front then lift it off the rest of the way. To replace the cover, just slide the tabs in at the front and lower it into place. Figure 1-2 shows the proper posi-

tion and movement for both removing and replacing the cover.

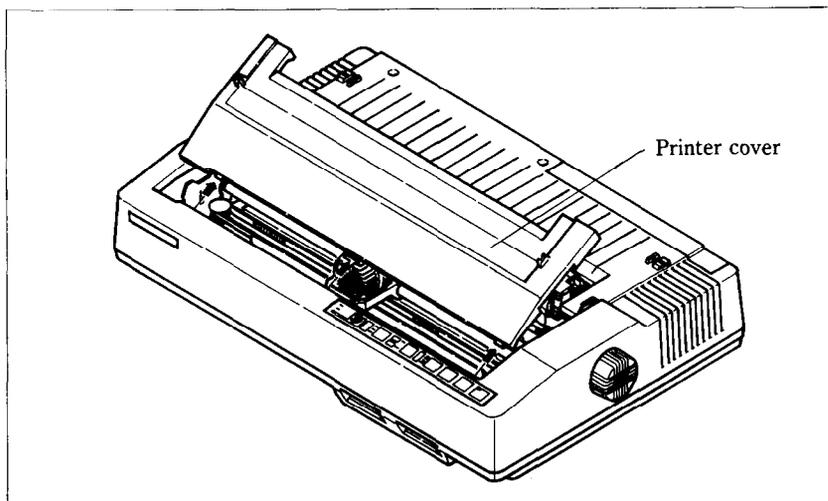


Figure 1-2. Remove the printer cover by lifting carefully.

■ Removing the packing tube

The printer is shipped with a protective spiral tube to keep the print head from being damaged during shipping. We have to remove this tube. First, take off the printer cover. See the tube on the carriage rail (Figure 1-3). Pull it off carefully.

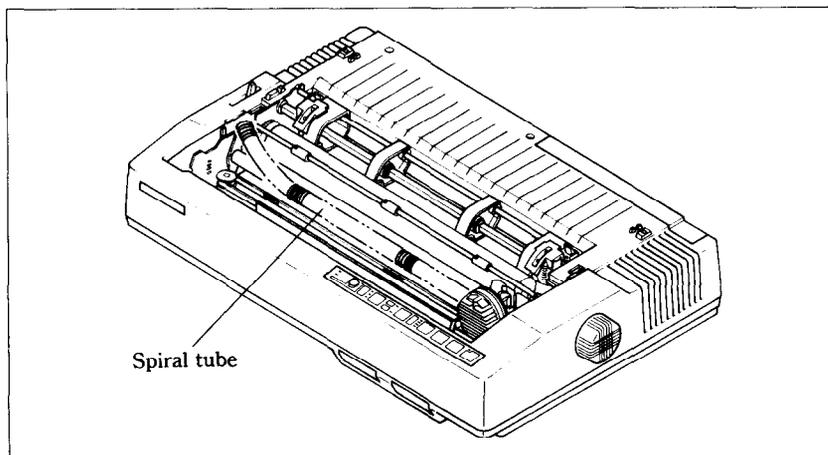


Figure 1-3. Remove the protective tube from the carriage rail.

■ Removing shipping screws

There are two shipping screws inside the printer, used to hold the internal chassis securely to the lower case during shipping.

To get at these, remove the printer cover from the printer. Then remove the screws with a Phillips screwdriver as shown in Figure 1-4.

You'll be smart to save these screws, along with the rest of the packing material and the shipping carton, in case you ever have to ship the printer. Tape the screws somewhere on the carton or packing.

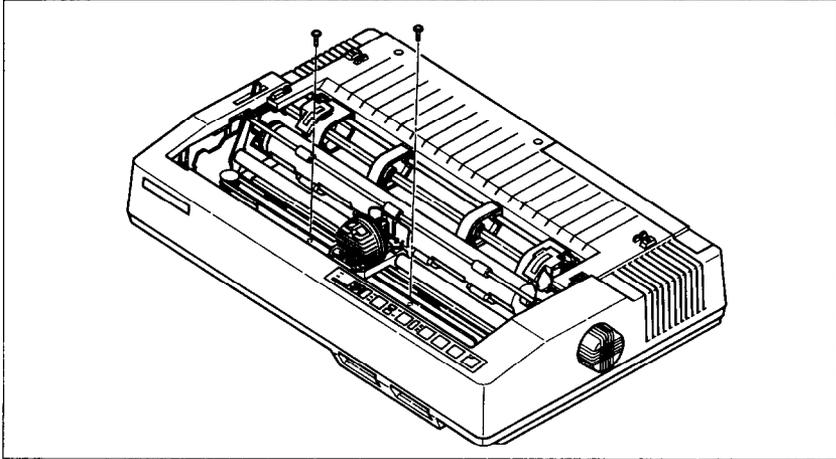


Figure 1-4. There are two shipping screws inside the printer, during shipping. They should be removed before use.

■ Installing the platen knob

This is the knob that turns the rubber platen cylinder. It fits into the hole in the right side of the printer case. Look at the hole in the knob. See the flat side of the hole? Match the flat surface in the knob with the flat side of the shaft, and press the knob onto the shaft firmly. Turn the knob a few times to make sure that the platen rotates smoothly and easily.

■ Installing the ribbon cartridge

This printer uses a neat, easy-to-change ribbon cartridge so you don't have to spend a lot of time threading a ribbon. And getting your hands all dirty to boot.

Telling you how to put in a ribbon is like explaining how to tie your shoelaces — it takes a lot longer to tell than to do. You can just follow the illustrations if you wish; they'll tell you all you really need to know.

Or, if you feel better following written instructions, read on.

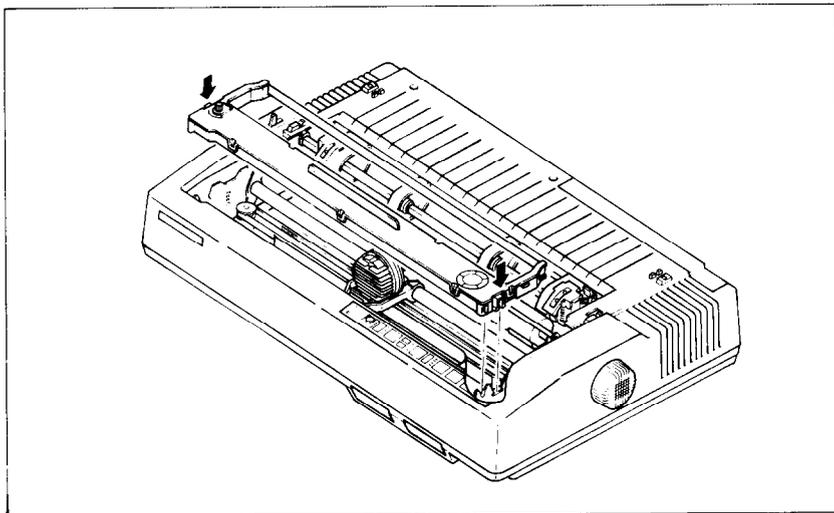


Figure 1-5. Press the cartridge into place until the holding springs snap into place.

1. Turn *off* the power and remove the printer cover.
2. Now slide the print head gently to the center of the printer.

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

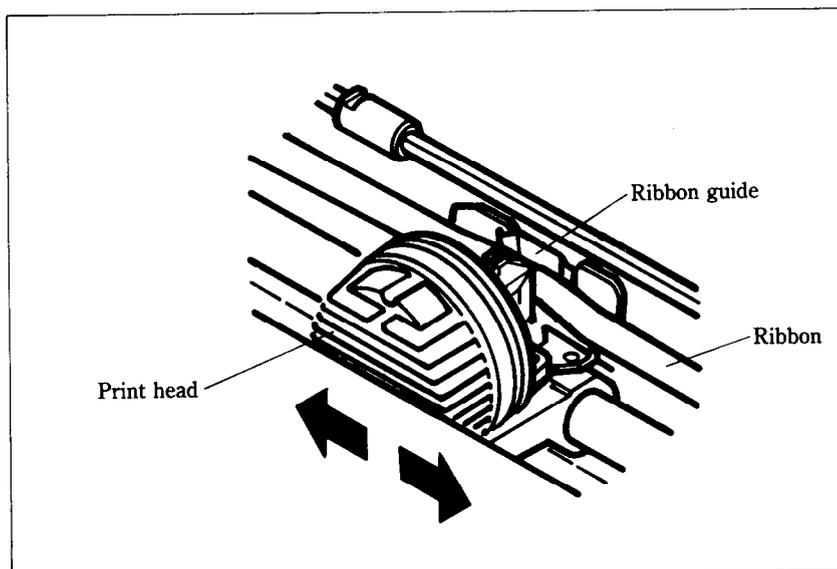


Figure 1-6. All you have to do is to slide the print head carriage manually to your right or left, then the ribbon slips down by itself into its proper position.

3. Using the guide holders as a fulcrum with the ribbon facing away from you, as shown in Figure 1-5, lightly press the cartridge down until the two holder springs snap shut to hold the cartridge firmly in place.
4. Check that the cartridge fits so that the drive pins engage the cartridge teeth.
5. Gently slide the print head carriage manually all the way to your right or left until the ribbon automatically slips down into its proper place between the print head and the silver ribbon guide.
6. Put the printer cover back on and you're finished. (A special switch on the printer prevents the printer from working when the cover is off.)

■ Installing the mute cover

The mute cover reduces the printer's operating sounds. Mounting the mute cover onto the printer cover is easy.

Note the position of the guide pins on the mute cover as shown in Figure 1-7. Then hold the mute cover at each end, and insert the guide pins into the cutout hooks of the printer cover. Finally, lay down the free-end of the mute cover backwards.

To remove it, just reverse the procedure.

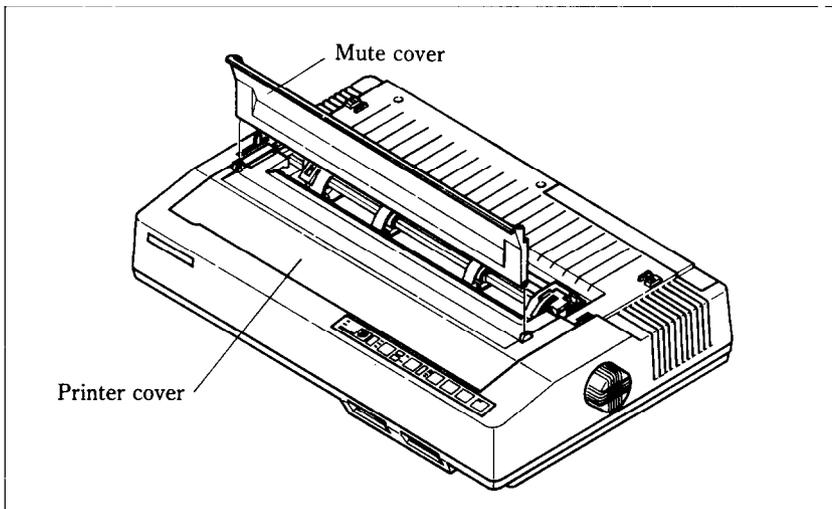


Figure 1-7. Install the mute cover by holding vertically, then lay down on the printer cover.

Up to this point, we've been clearing the decks for action, so to speak. There are only a couple more things left to do before we can start printing — install the interface board and the font cartridge.

■ Installing the interface board

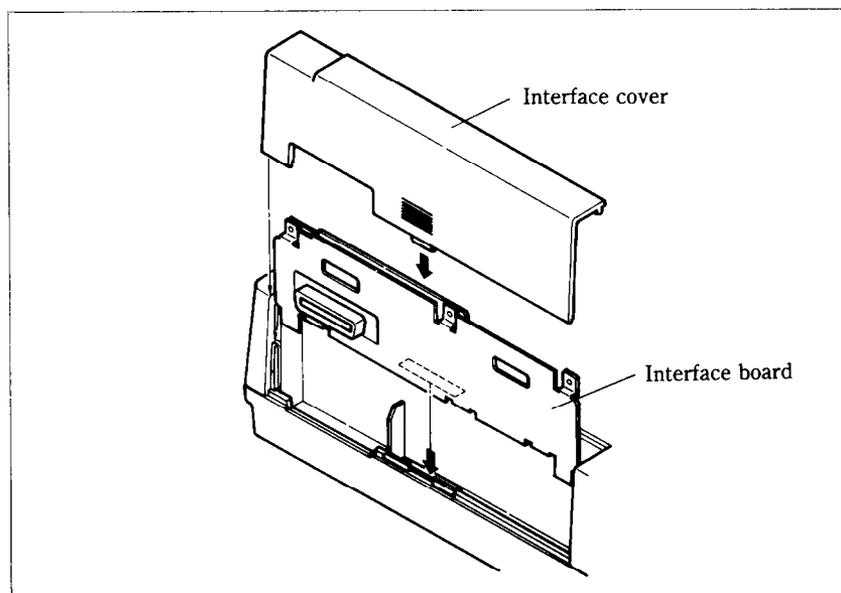


Figure 1-8. Slide the I/F board into the slot.

The parallel interface board that comes with your printer contains all the electronics that your printer needs to talk to a computer. In addition, there are two optional interface (I/F) boards that you can order: the serial interface board lets you connect to a serial (RS-232C) computer port; the other board contains extra buffer memory for parallel output. All three boards are installed the same way.

To install or change an I/F board, start by turning the printer around. Notice the snap-on cover at the left end. Take it off and you'll see where the board goes.

Fit the I/F board into the side slots and slide it all the way down (don't force it) by turning off the power switch. If the board connector is seated snugly in its socket, you should be able to put the cover back on and connect the cable from your computer— you're ready to go.

■ Installing the font cartridge

This printer is mounted the Prestige characters as standard. You can add two different character styles by installing the optional font cartridges to the printer.

To install or change a font cartridge, start by turning off the power switch.

Fit the font cartridge into the slots and slide it all the way.

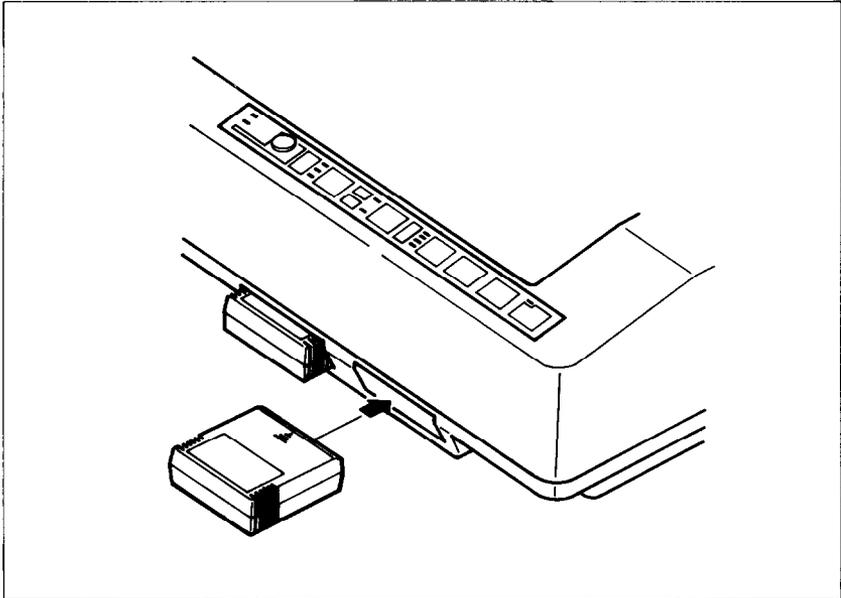


Figure 1-9. Slide the font cartridge into front slot while the power switch off.

CHAPTER 2

GETTING TO KNOW YOUR PRINTER

Subjects we'll cover in Chapter 2 include —

- **Parts of the printer** — what they're for and how to use them;
- **Paper selection and loading;**
- **Adjustment;**
- **Testing printer operation.**

CONTROLS AND PARTS OF THE PRINTER

■ Parts of the printer

First, we'll go over the parts of the printer. You saw most of these when you unpacked your printer. Now we'll give you a brief explanation of what they do.

PRINTER COVER — This protects the ribbon and the print head from dust and dirt, and cuts down the sound of the printer.

MUTE COVER — This reduces the sound of the printer by installing on the printer cover.

PAPER GUIDE — As you've guessed, this flat plastic molding guides the paper during printing (it is raised for single sheets and lies flat for sprocket-feed paper).

POWER CORD — This cord connects the printer to its power source, usually a wall outlet. It's located at the left rear of the printer.

PRINT HEAD — This is the device that does the actual printing. Like the strike lever in a typewriter, tiny stiff wires in the print head hit the paper through a ribbon.

TRACTOR FEED UNIT — The drive gear and sprockets of the tractor feed unit move sprocket-feed paper through the printer.

PLATEN — This is the rubber cylinder that carries paper to the print head.

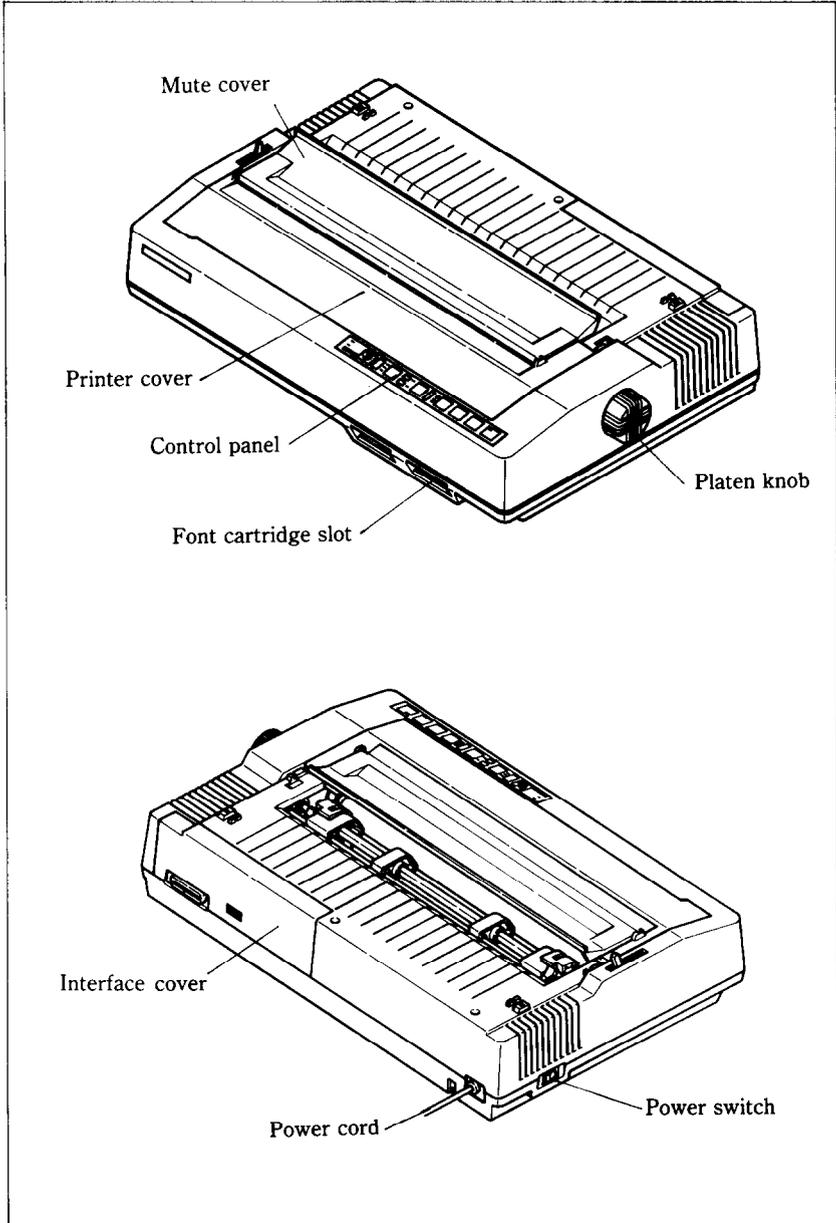


Figure 2-1. Front and rear views of the printer.

INTERFACE COVER — On the back of the printer, this conceals and protects the interface board. The cover is cut away to expose the external connector to which you attach your computer cable.

SLOTS FOR FONT CARTRIDGES — These are the two openings at the bottom right of the front of the printer. Inserting an optional font cartridge pushes the small door up out of the way and exposes a connector which accepts the cartridge.

■ Controls and indicators

Now let's go over the controls and indicators of the printer, starting with the control panel on the right. There are seven indicators and five keys on the control panel.

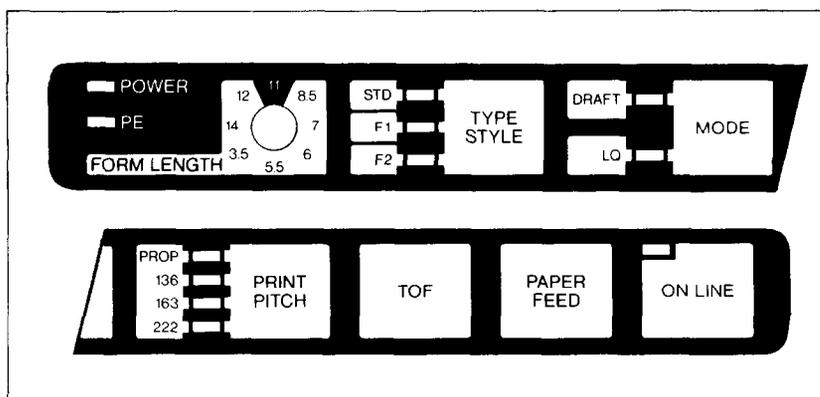


Figure 2-2. The control panel.

POWER INDICATOR — Glows green when the power is *on*.

PE INDICATOR — Stands for “Paper Empty”, and glows red when the printer runs out of paper and when certain printer malfunctions occur.

FORM LENGTH SWITCH — Tells the printer how long the pages are so that it can generate an automatic TOF code at the appropriate time.

TYPE STYLE INDICATORS — Shows the character set in use: standard (STD), font 1 (F1), or font 2 (F2).

TYPE STYLE KEY — Selects one of the three character sets available to the printer.

MODE INDICATORS — Either DRAFT or LQ glows green to indicate the mode (set by the Mode key or the DIP switch).

MODE KEY — Changes the print mode every time it is pressed.

PITCH INDICATORS — Shows the character pitch currently set: Proportional (PROP), 136, 163, or 222 characters/line.

PRINT PITCH KEY — Selects one of the four available character pitches.

TOF KEY — Ejects the current sheet, feeds a new sheet, and moves the print head to the left margin.

PAPER FEED KEY — Advances the paper one line at a time when the On Line indicator is off. If you hold the key down, you can get consecutive line feeds, one after the other.

ON LINE KEY and INDICATOR — Glows green when the printer is ready to receive data from your computer (on line). When the printer is off line, it sends a signal to the computer indicating that it cannot accept data. Pressing the On Line key switches the printer on line and off line. When you turn the power switch on, the printer is placed on line automatically.

■ Extra functions

There are twelve more functions that are not directly specified on the control panel. You can use seven of them by pressing a key when you turn on the power; five others are activated by pressing two control keys at once.

For details, please refer to the end of this chapter.

■ Other controls

There are other controls, not connected to the control panel. Some of the more important ones are:

POWER SWITCH — The switch that turns the printer on and off is at the back, on the left side.

PLATEN KNOB — This knob is at the middle of the right side and lets you turn the platen by hand, just like a typewriter.

Caution: Turn the platen knob only with the Power switch *off*.

Turning the knob with the power on could damage the drive gears.

RELEASE LEVER — The release lever is on top of the printer, near the left rear corner. You'll be using it often — it controls how strongly the paper is held against the platen. The release lever has four positions: the top setting is used for inserting paper, the next is for single sheets, the third is for sprocket-feed paper, and the bottom one is used when adjusting the paper.

PAPER BAIL — The bail is the movable bar that holds the paper against the platen. Its position depends on the setting of the release lever.

DIP SWITCHES — There are two sets of eight switches that are used in interfacing the printer to your particular computer. There are also switches to set page length, line spacing, and per-

foration skipping. See Appendix A for a complete list and explanation.

SELECTING AND LOADING PAPER

Your printer can handle the two basic types of paper — single sheets (stationery, envelopes, multipart business forms, etc.) and continuous paper (fan-fold perforated paper).

This is a good place to tell you about the release lever, which you'll be using often. This lever controls the pressure of the paper against the platen.

It has four settings:

-  — The top setting is used when you want to load paper in the printer.
-  — The second setting is for normal single sheets.
-  — The third is for sprocket-feed paper.
-  — The bottom one is used when you want to release the paper completely to adjust it.

The paper bail holds the paper against the platen according to the setting of the release lever. The bail is opened when the release lever is in the top position, closed when the lever is in the second or third positions, and opened by the printer when the lever is in the bottom position.

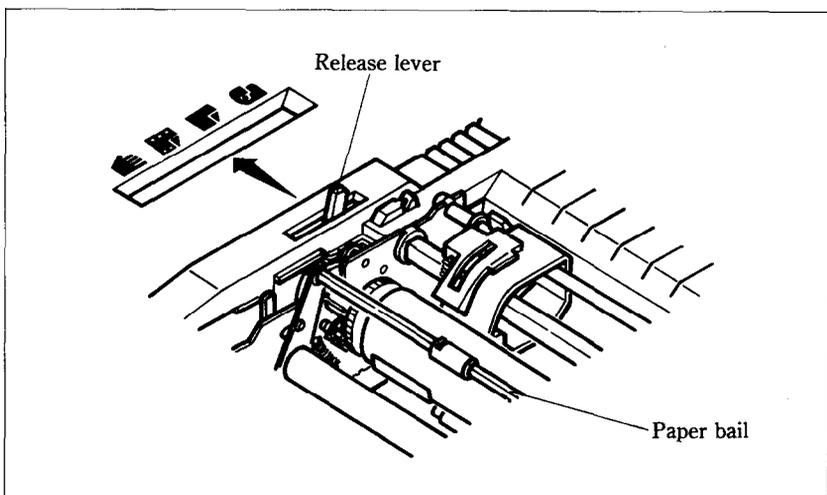


Figure 2-3. Use friction feed for single sheets and tractor feed for continuous paper.

■ Loading single sheets

Now, instead of feeding the paper in *manually* by turning the platen knob, we're going to use the release lever for automatic feeding (the printer must be turned *on*). Single sheets from 5.5 to 14.5 inches wide can be used.

OK? Let's start.

1. Move the sprockets and sheet guides on the tractor feed unit all the way to the ends.
2. With the ribs of the paper guide toward you and the sliding adjuster at the bottom left, insert the guide into the slot at the back of the printer cover.
3. Insert the guide stand in the notches provided for it and use it to prop up the paper guide.
4. Position the adjuster at the triangle marked on the guide for standard margins, or set it according to your requirements.
5. Set the release lever for single sheets—the second position from the top.

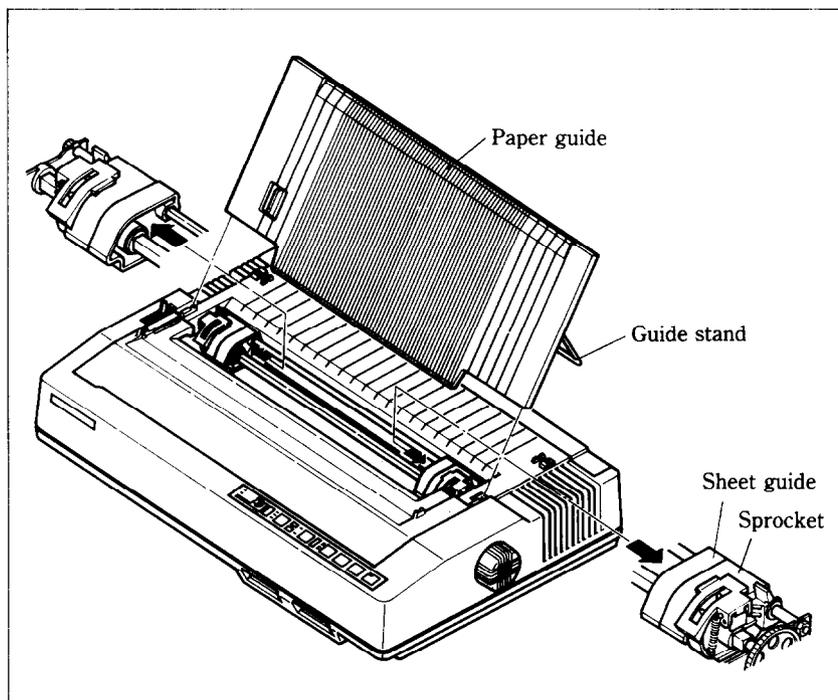


Figure 2-4. Raise the paper guide for loading single sheets.

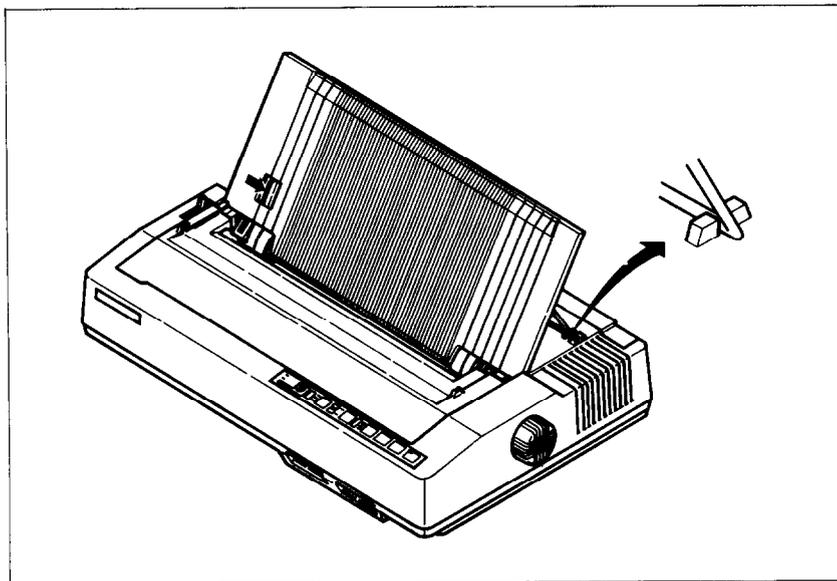


Figure 2-5. Prop up the paper guide by inserting the guide stand in the notches provided for it.

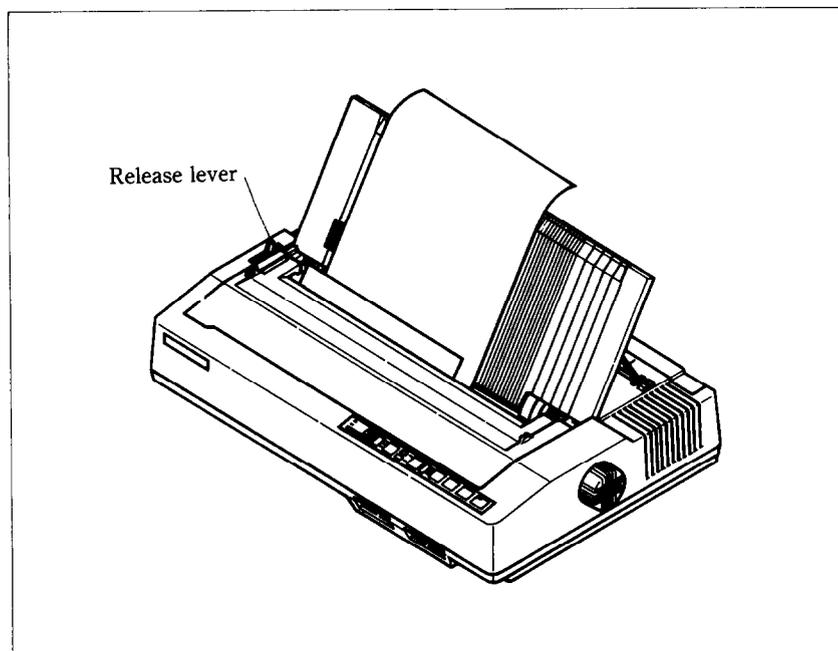


Figure 2-6. You can insert a sheet of paper by using the release lever.

6. Align a sheet of paper on the guide along the adjuster.
7. Turn on the power switch. (The Paper Empty indicator will glow because there is no paper.)
8. Now set the release lever to the auto-feed setting—the top position. The printer will advance the paper automatically.
9. When the paper stops, set the release lever for single sheets again.

To align paper that is not in straight—

- Set the release lever to the adjustment setting.
- Straighten the paper and adjust it for the margins you want.
- Move the release lever back to its original setting.

Remember, you can also use the optional automatic sheet feeder for single sheets.

■ **Loading sprocket-feed paper**

This is the familiar computer paper, with the holes along the sides and perforations between the sheets. It is also called sprocket, punched, fan-fold, or just plain “computer paper.” It can be as narrow as 4”, and up to 15.5” wide.

1. Turn on the printer and set the release lever for sprocket-feed paper.
2. If the paper guide is installed, remove it.
3. Place a stack of fan-fold paper behind the printer.
4. Open the sprocket covers, on the right and left sprocket units, as shown in Figure 2-7.
5. Flip the clamp levers backward. This allows the two sprocket units to move freely right and left so you can align them with the holes in the paper.
6. Feed the end of the paper into the slot next to the platen cover plate.
7. Fit the holes in the paper over the sprocket pins so the paper is even, and clamp the sprocket units in place.
8. Check that the paper is still positioned correctly, and close the sprocket covers (Figure 2-8).
9. The Paper Empty indicator should be glown. Turn the platen knob to feed the paper until the indicator goes out.
10. Now turn the platen knob the other way just a bit, until the indicator glows again.

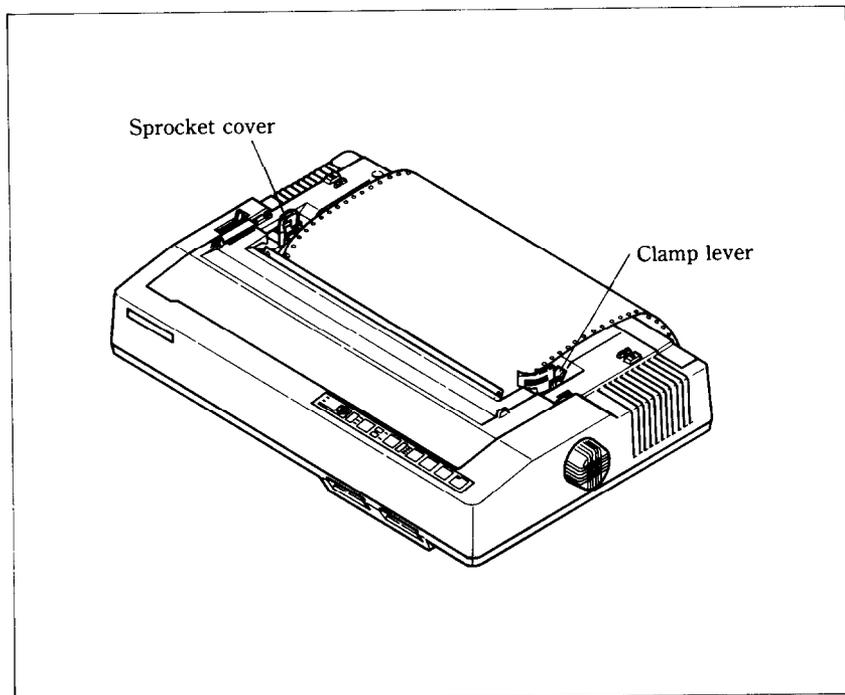


Figure 2-7. Open the sprocket covers to expose the sprocket teeth.

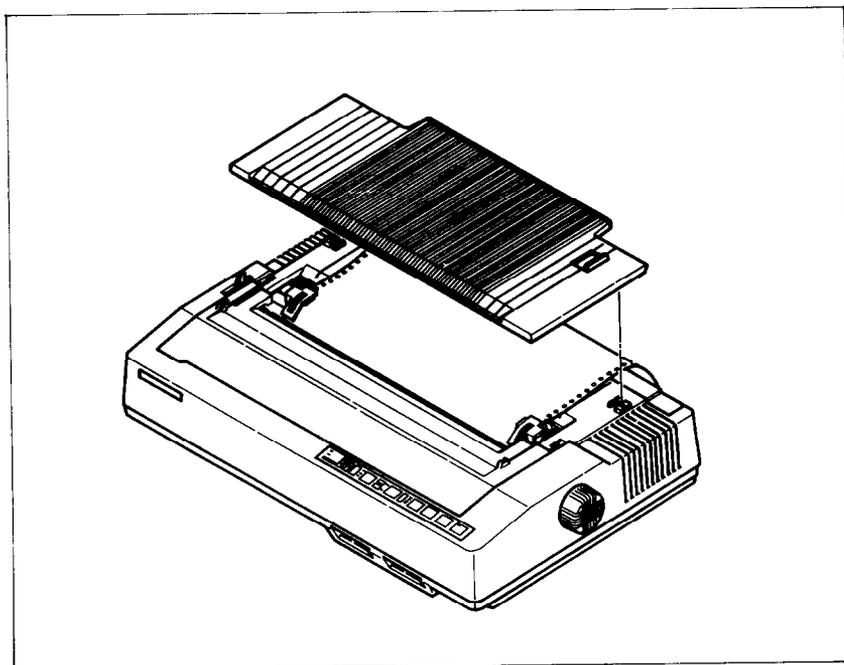


Figure 2-8. Ready to run with sprocket-feed paper.

11. Set the release lever to the auto-feed setting. The printer will advance the paper automatically.
12. When the paper stops, set the release lever for sprocket-feed paper again.
13. Finally, install the paper guide on the printer. You can install it either to lie flat as shown in Figure 2-8, or to raise as shown in Figure 2-9. Now you're ready to roll!

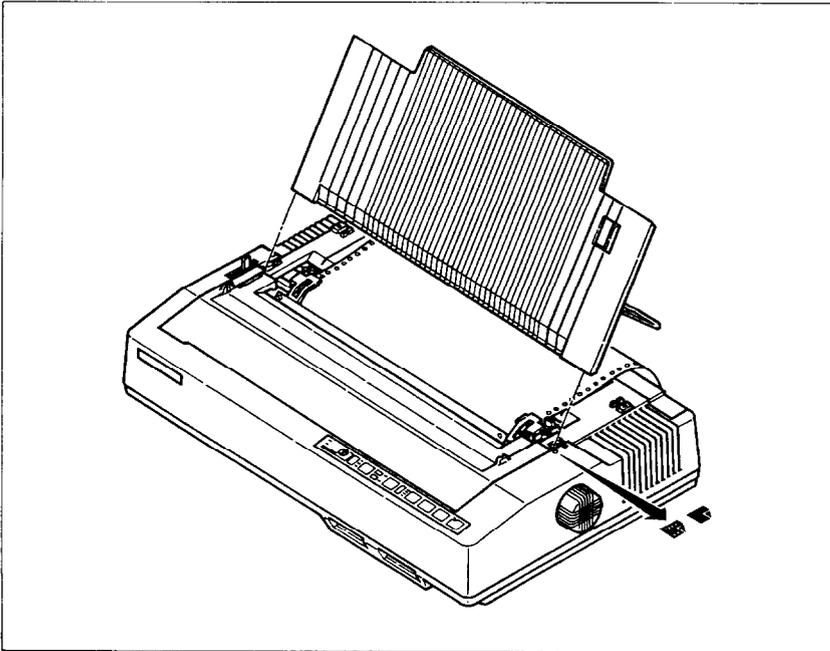


Figure 2-9. You can raise the paper guide with sprocket-feed paper.

ADJUSTING THE PRINT HEAD

The distance between the print head and the platen must be adjusted to accommodate papers of different thicknesses. To make this adjustment, move the adjustment lever, which is under the printer cover and immediately in front of the release lever (Figure 2-10). Pulling the adjustment lever towards you will widen the gap; pushing it away from you will narrow the gap.

There are four positions; you can feel the lever clicking into the various notches. The first step (shown) is the one most commonly used for single sheets of paper.

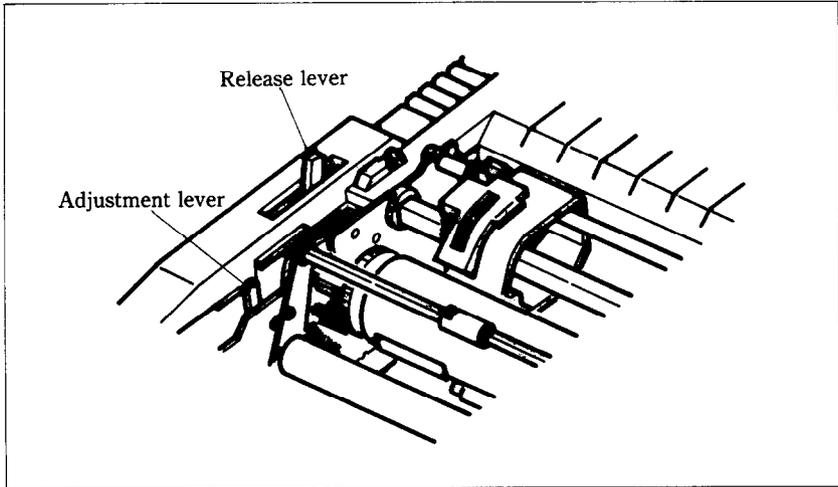


Figure 2-10. The adjustment lever allows for different thicknesses of paper.

You shouldn't have any trouble finding the right gap setting for your paper. If necessary, experiment; you'll soon find the best position for the paper you're using.

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.

While turning on the power —

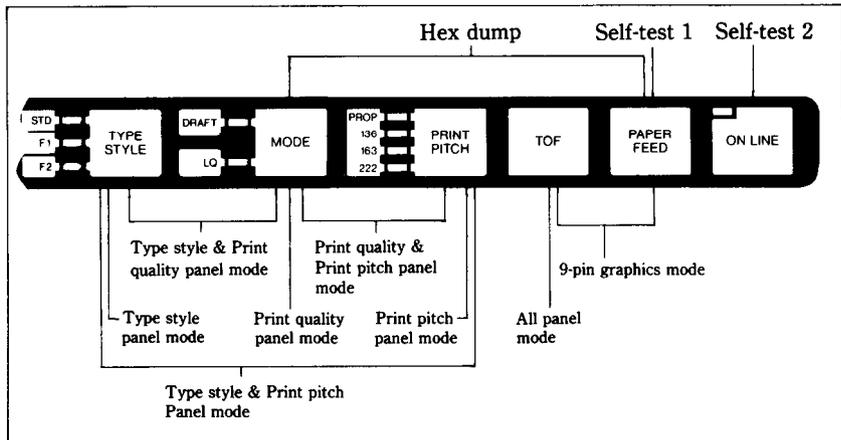


Figure 2-11. Extra functions while turning on the printer.

1. Plug in the printer (don't turn it on yet).
2. Insert a sheet of paper, as you did for the self-tests.
3. While holding down both the Paper Feed and Mode keys, turn on the power switch.

When you want to terminate the hex dump function, turn off the power switch.

■ Panel modes

As you'll learn in Chapter 5, this printer has many software controls. But if you want to print in one fixed mode, ignoring the control codes, these Panel modes take effect for you. To set the Panel modes, follow the procedures:

1. Plug in the printer (don't turn it on yet).
2. While holding the key or keys as shown below, turn on the power switch.

<u>Key</u>	<u>Type of Panel mode</u>
Type Style	Character set type
Mode	Print quality (draft or LQ)
Print pitch	Print pitch
TOF	All of the above

Notice that these modes stay on with ignoring the relative control codes until you turn off the printer.

■ 9-pin graphics emulation mode

You already know that this printer represents the state of the art in tack-sharp 24-pin text printing. To complete the text ability of this printer, we have also the ability to do dot graphics 9-pin printing. The IBM graphics printer mode utilizes 9-pin graphics software which is commonly used by the IBM printer.

The IBM printer requires a different line feed pitch of either $n/72$ nd or $n/216$ th inches. This printer normally uses a pitch of $n/180$ th inches, due to its 24-pin design. To avoid software incompatibility with existing IBM graphics programs, this printer has been designed to most closely emulate the requirements for 9-pin dot graphics compatibility.

This *unique* 9-pin graphics emulation mode can be evoked as follows:

1. Plug in the printer (don't turn it on yet).
2. While holding the TOF key and the Paper Feed key, turn on the power switch.

When you want to cancel this mode, turn off the power switch.

During normal operation —

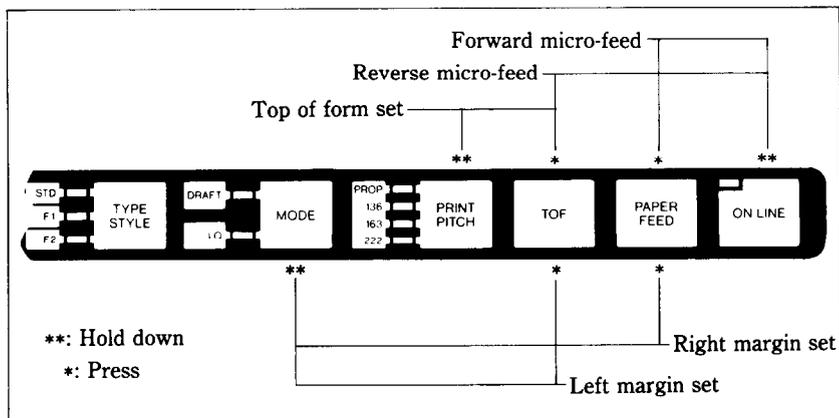


Figure 2-13. You can set many functions by the combinations of the control panel keys while in the Off Line mode.

■ 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 reset it by the following procedures.

1. Set the Off Line mode by pressing the On Line key.
2. While holding the Print Pitch 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 5, you can set the left and right margins with the control codes. In addition, you can set them manually just like a typewriter 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.

CONNECTING THE PRINTER

Now that you've assembled and tested your printer, you're surely looking forward to using it forward what you bought it for — printing! We're almost there — just connect it to your computer as follows:

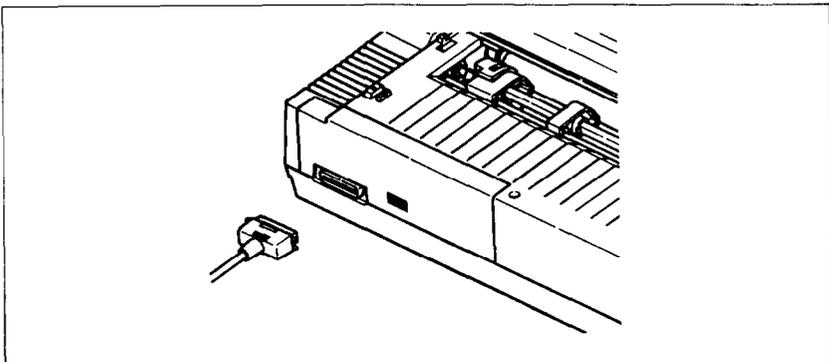


Figure 2-14. Connect the cable like this.

1. Make sure that both your computer and printer are turned off.
2. Connect the printer end of the interface cable to the connector socket at the right rear of the printer as shown in Figure 2-14.
3. Connect the other end of the cable to your computer as described in the computer manual.

MEMO

CHAPTER 3

USING COMMERCIAL SOFTWARE

Subjects we'll cover in Chapter 3 include—

- Using your printer with SuperCalc;
- Using your printer with word processors.

Many of you purchased this printer to use with commercial software. You made a good choice because this printer is compatible with most commercial program, from word processing programs to spreadsheet programs to accounting programs.

Many of these programs have a routine for describing your printer. These routines are often in "installation programs." They typically give you a choice of printers or printer types to pick from. Some typical descriptions that you might pick for your printer are: "TTY type printer with backspace," "IBM-dot matrix printer," "Centronics-type printer," "Dot matrix ASCII printer." Your printer should work fine with any of these descriptions.

Some printer lists are not very clear, and may not include anything that you think describes your printer. If you can't decide which description best fits your printer we recommend that you narrow the list to two or three choices (you can quickly eliminate all the daisywheel printer types) and then experiment. You won't hurt anything if you guess wrong; it just won't work right. This should quickly tell you if your guess is right. If all else fails, though, your printer dealer will be happy to give you some advice.

Some programs don't ask you what kind of printer you have, but instead they ask some questions about what your printer can do. Here are the answers to the "most asked" questions. Your printer *can* do a "backspace". Your printer *can* do a "hardware form feed".

With the questions answered, you are ready to start printing. Read the manual that came with your commercial software to see how to make it send for your printer to print. This is all you

need to know to use your printer as a regular printer. But this printer isn't just a regular printer. Your printer has many capabilities that your commercial software isn't aware of. A little later we will see what it takes to use some of your printer's advanced features with commercial software.

■ First, some terminology

Your printer knows what to print because it knows how to interpret the codes that the computer sends to it. These codes are numbers that the computer sends to your printer. Both the computer and the printer know the meaning of these codes because they are a set of standard codes used by almost all microcomputers. This set of codes is the *American Standard Code for Information Interchange*, which is usually referred to as ASCII (pronounced *ask-key*). There are ASCII codes for all the letters of the alphabet, both lower case and capital, the numbers from 0 to 9, most punctuation marks, and some (but not all) of the printer's functions.

ASCII codes are referred to in several different ways, depending on the way they are used. Sometimes these codes are treated as regular numbers. For example, the letter "A" is represented by the number 65 in ASCII. Appendix B shows all of the ASCII codes.

In BASIC, ASCII codes are used in the CHR\$ function. This function is used to print the character that is represented by the number in the CHR\$() function. The BASIC statement PRINT CHR\$(65) will print an "A" on the terminal.

In some other programming languages, ASCII codes are referred to by their *hex* value. "Hex" is short for *hexadecimal* which is a base-16 number system (our usual numbers are base-10). Since hex needs 16 digits, it uses the numbers 0 through 9 and then it uses the letters A through F for digits. The ASCII code for the letter "A" is 41 in hex.

Of course, most of the time we don't even need to think about this code system. Our computers are smart enough to know that when we press the "A" key on our keyboard we want to print the letter "A". The computer takes care of all the rest.

But there are a number of ASCII codes that don't have keys on the keyboard. The most important of these codes are the codes that ASCII values below 32. These codes control many of this printer's functions. Even though there aren't keys for these codes, most keyboards can send these codes. It's done by holding down the "control" key (many times marked CTRL) and

simultaneously pressing a letter key. The particular letter key that is pressed determines what code is sent. Control and A sends ASCII code 1, control and B sends ASCII code 2, and so on. Because of the way they are created, these codes are often referred to as “control-A” etc.

So there are four common ways of referring to the same set of codes: the character or name of the code, the decimal ASCII value, the hexadecimal ASCII value, and the “control-” value.

For example, the code that causes the printer to advance the paper one line is ASCII 10 (decimal). This code is commonly referred to by all the following names:

Line feed	— its name
<LF>	— the abbreviation of its name
ASCII 10	— its decimal value
ASCII &H0A	— its hexadecimal value (the &H signifies hex)
CHR\$(10)	— the way it's used in BASIC
control-J	— the way you send it from a keyboard

There's a chart in Appendix B that shows these side-by-side so that you can convert back and forth.

The reason that we are telling you all this about ASCII codes is that people are not very consistent about how they describe ASCII codes. We are going to help you use your printer with commercial software, but we don't know what its documentation is going to call the various codes. So if you know all the different things that the codes might be called, it will be easier to figure out what it is trying to tell you.

Now, armed with the knowledge of what to look for, you can delve into manuals of your commercial software and dig out the secrets of how to send “control codes” to your printer. When you find the method that your program uses, then you can shop through this manual to find the function that you want to use. By translating the codes from the system that we use, to the system that your commercial software uses, you should be able to use many of your printer's advanced features. It may help, however, if we look at a couple of examples.

USING YOUR PRINTER WITH SUPERCALC

SuperCalc is typical of the many spreadsheet programs that

are now available. It has the capability of using several of the advanced features of your printer. Perhaps the most often used feature with spreadsheet programs is compressed printing. Let's see how to use compressed printing with SuperCalc.

In SuperCalc, the /Output command provides output to the printer. One of the options of the /Output command is S(etup). This option provides you with a menu of functions to configure SuperCalc to match your printer. You can change the number of characters that SuperCalc will print on a line and the number of lines that will print on a page. You should be sure that these values match your printer. Your printer can print 136 characters per line of pica type, or 234 characters per line of condensed elite type. One of the other options on this menu is "send setup codes to printer". This is how we tell this printer that we want to use condensed print. The code to switch this printer into condensed print is ASCII 15, or control-O. So to switch on condensed type, use the /Output command and, after selecting D(isplay) and entering the range to print, select the S(etup) option, and the S(etup)-"Manual setup codes" sub option. Then, at the prompt that says "Enter codes (CR when done)", type control-O. Remember, to enter control-O you should hold down the CTRL key while you press the O key (That's the letter Oh, not the number zero). Then just press the return and select P(rint) to print your report.

You only need to go through this procedure once each time you use SuperCalc because your printer will stay in compressed print until it's turned off or reset.

You might also wish to use some of your printer's other features with SuperCalc. Find the code for the feature you wish to use in Appendix C and use the same procedure given here.

USING YOUR PRINTER WITH WORD PROCESSORS

Not many word processing programs recognize the advanced features of printers like your printer. They usually provide for some method of making bold characters and underlining. But your printer can do much more than that. The people that write word processing programs do, however, know that there are a lot of different printers on the market, and so they usually, (but not always) provide a way of sending special codes to a printer. We will study one example of this to see how a typical word pro-

cessor handles it. Once you understand the concept you should be able to use your program manual to figure out how your word processor does it.

The program that we will study is the EasyWriter word processor for the IBM Personal Computer. This uses a fairly typical method of handling special codes. Generally, word processing programs don't want you to put non-printing codes in the file. They "know" that they won't print anything, and so they "protect" you by not letting you use them. But the non-printing codes are the ones that you need to use your printer's features. So EasyWriter provides a way to override this protection. If you precede a special code with a "control-O" then EasyWriter will accept the next non-printing code.

Let's look at a specific example. Suppose you want to print the title of a book in boldface. The code sequence to select boldface print is Escape G (that's two separate characters). Entering the G is no problem; it's a printing character so EasyWriter won't object (although in this case it's not going to print). The Escape, however, is a non-printing character so it requires special handling. To enter the Escape code first enter control-O (hold the Ctrl key while you press the letter O). Then press the Esc key. The Escape character shows on the screen as a left pointing arrow. Now just type the letter G and you're done.

When you want to end the boldface print, you need to enter Escape H. Use the same procedure: enter control-O, Esc, and then H.

You can use many of your printer's features this way. Find the codes that you need in Appendix C.

A note to WordStar users: WordStar is probably the most popular word processing program in the world. But it provides no way to enter special printer control codes from the keyboard. WordStar does, however, provide you with a way to use some of your printer's advanced features. WordStar has four special commands that you can use to access your printer's features. These are called "user printer controls" and are control-P Q, control-P W, control-P E, and control-P R. You might use two of these to turn boldface on and off and the other two for some other function. The process of setting up these codes is called "patching" and is done with the install program that comes with WordStar. The procedure is fairly involved, but it is explained in the WordStar manual. If you have trouble figuring it out, ask for assistance where you bought WordStar.

MEMO

CHAPTER 4

CONTROLLING WITH BASIC

Subjects we'll cover in Chapter 4 include—

- Listing BASIC programs on the printer;
- Printing from BASIC;
- CHR\$ function;
- Command syntax;
- Problem codes.

In this chapter we will be teaching you how to use your printer's features using the BASIC programming language. It is easy to communicate with your printer from BASIC and, though it has its detractors, BASIC is the nearest thing to a universal language among users of personal computers. But remember that it's not the only way to communicate with your printer, as we have already seen.

SOME BASICS ABOUT BASIC

■ 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 Microsoft 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 prefix the command with an L (enter LLIST).

The Microsoft BASIC command for outputting information is PRINT. Like the LIST command, this displays the information on the CRT screen so we have to add an L (→ LPRINT) if we want to use the printer. Just put whatever you want to print between quotes and after LPRINT (anything enclosed in quotes is called a character string). For example, we would use LPRINT "Hello!" to output "Hello!" to the printer. We'll see later how to

LPRINT things other than character strings.

We started with Microsoft BASIC because it is the most widely used version of BASIC around. The programs in this manual are written in Microsoft BASIC so they should run on most computers. But if strange things happen when you try to run a program, check the BASIC manual that came with your computer.

Let's talk about Apple II computers for a minute. These enormously popular computers use their own brand of BASIC. To use an Apple II, enter the following —

```
PR#1          PR#1
LIST          PRINT "Hello!"
PR#0          PR#0
```

The PR#1 tells the Apple to send everything to the printer, the LIST or PRINT command sends it, and the PR#0 returns output to the screen.

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 "Hello!" or to list a program in the C-64's memory.

```
OPEN4,4       OPEN4,4
CMD4          CMD4
LIST          PRINT "Hello!"
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.

Appendix H gives more information about listing programs on the various computers. Find the part that applies to your computer and try it.

Now that we all know how our computers address the printer, let's try listing a BASIC program. Load a BASIC program and LLIST it (or however your computer does it). We've crossed the first major hurdle—learning how to list programs on your printer. Now we are ready to jump into the world of programming with your printer. But first, there are a few fundamentals that we need to cover.

■ Establishing communications

We've learned something about communicating with our printer. Now we need to adapt what we know to printing in a

BASIC program. Generally, computers use about the same procedure for printing in a program as they do to list a program.

Let's try what we learned. Type the following:

```
NEW
10 LPRINT "TESTING"
RUN
```

Remember—we use LPRINT; you may have to use something else!

At any rate, you should have the word "TESTING" on your printer. Quite an achievement, isn't it? Let's get done with this simple stuff so that we can go on to something interesting.

■ The CHR\$ function

We mentioned CHR\$ in Chapter 3 as one way to express ASCII codes. We are going to use it a lot in communicating with this printer. This printer uses many of the ASCII code that don't represent letters and numbers. The CHR\$ function gives us an easy way to send these codes to the printer. Try this to see how the CHR\$ function works:

```
NEW
10 LPRINT CHR$(80)
RUN
```

That should print a "P" for Printer. If you check the chart in Appendix B you will see that 80 is the ASCII code for "P".

■ Control codes

This printer uses many of the non-printing ASCII codes for control codes. These codes perform a function rather than printing a character. Let's try an easy one right now:

```
NEW
10 LPRINT CHR$(7)
RUN
```

Where did that noise come from? That's the printer's bell. We will learn more about it in Chapter 5. We just wanted to illustrate a code that causes the printer to perform a function.

■ 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> "W" 1
```

This command that turns on expanded printing. `<ESC>`, as we mentioned earlier, is the escape code (which is ASCII code 27). A letter or number in quotes (such as the "W" above) means that the character should be sent to the printer (without the quotes). In our example, you should send a capital W following the escape code. In BASIC, you could do this in a couple of ways: by sending the character itself (e.g. `LPRINT "W";`), or by using the `CHR$` function to send the ASCII code for the character (e.g. `LPRINT CHR$(87);`).

Many of printer commands end with a 1 or 0. When shown as in the above example (i.e. no quotes and no "CHR\$"), you can use either ASCII code 1 (i.e. `CHR$(1)`) or the character "1" (which is ASCII code 49). The same idea applies to commands

ending with 0.

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

```
LPRINT CHR$(27);"W"CHR$(1)
LPRINT CHR$(27);"W"CHR$(49)
LPRINT CHR$(27);"W1"
```

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.

■ Some problem codes

Before we go too far we need to mention some codes that may cause you problem.

Nearly all BASICs change some of the ASCII codes between your BASIC program and your printer. IBM's Microsoft BASIC adds a CHR\$(10) (a line feed) to a CHR\$(13) (a carriage return) before sending it on. Another problem code is ASCII 26. Once again we refer to Appendix H where some more specific information awaits.

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

MEMO

CHAPTER 5

PRINTING TEXT

Subjects we'll cover in Chapter 5 include—

- Letter quality (LQ) characters;
- Underlining and overlining;
- Superscripts and subscripts;
- Print boldface;
- Print pitches.

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.

All our examples will be given in Microsoft BASIC as used by the IBM Personal Computer, but remember that you don't need to know BASIC to use your printer's features. Just use the same ASCII codes as we do in our examples with all DIP switches are set on condition.

If your computer doesn't use Microsoft BASIC, look in the appendix to see what changes you need for your BASIC.

You have already printed a few lines on your printer. Now it's time to start looking at the many variations of printing style that you have available to you.

SOME SPECIAL KINDS OF TEXT

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

■ Letter Quality characters

This printer's Letter Quality (sometimes abbreviated as LQ) 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 when the DIP switch 1-4 is set off. This is adequate for most work and it prints fastest. But for the final printout, try Letter Quality. The program below shows how.

```
10 ' Demo of LQ character set
20 LPRINT CHR$(27);"x1";
30 LPRINT "This line shows LETTER QUALITY!"
40 LPRINT CHR$(27);"x0";
50 LPRINT "This line shows standard print."
```

In this program, line 20 selects LQ characters with $\langle \text{ESC} \rangle$ "x" 1 command. Line 30 prints a sample before line 40 switches printer back to draft printing with an $\langle \text{ESC} \rangle$ "x"0. When you run the program you should get this:

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

Table 5-1
Letter quality commands

Function	Control code
Letter quality ON	$\langle \text{ESC} \rangle$ "x"1
Letter quality OFF	$\langle \text{ESC} \rangle$ "x"0

■ Characters in the Font Cartridge

This printer offers you the Prestige LQ characters as standard. If you want to use other LQ character styles, use the optional font cartridges.

You can install those font cartridges following the procedures described in Chapter 1. You can control the characters in those font cartridges as shown below:

```
10 ' Demo of optional characters
20 LPRINT CHR$(27);"x1";
30 LPRINT CHR$(27);"k";CHR$(1);
40 LPRINT "This line shows Font 1 characters."
50 LPRINT CHR$(27);"k";CHR$(2);
```

```

60 LPRINT "This line shows Font 2 characters."
70 LPRINT CHR$(27);"k";CHR$(0);
80 LPRINT "This line shows Internal Characters."
90 END

```

When you've installed the Gothic Font Cartridge to the font slot 1, and the Italic Prestige Font Cartridge to the font slot 2, the results of this program look like this:

```

This line shows Font 1 characters.
This line shows Font 2 characters.
This line shows Internal Characters.

```

In this program, line 30 selects the characters on the font slot 1, with `<ESC> "k" CHR$(1)` command. Line 40 prints a sample with the LQ characters in the font cartridge installed on the font slot 1. Line 50 selects the font cartridge slot 2 with `<ESC> "k" CHR$(2)` command. Line 60 prints a sample with the font slot 2 characters before line 70 switches printer back to internal LQ characters with an `<ESC> "k" CHR$(0)` command.

You can control the font styles with the extra functions with the control panel.

Table 5-2
Font selection commands

Function	Control code
Selects font slot 1	<code><ESC> "k" CHR\$(1)</code> or <code><ESC> "4"</code>
Selects font slot 2	<code><ESC> "k" CHR\$(2)</code>
Selects internal characters	<code><ESC> "k" CHR\$(0)</code>
Cancels font slot 1	<code><ESC> "5"</code>

■ Underlines and overlines

Not only your printer print all characters in both draft and LQ, but it can underline or overline them too. The control codes are shown in Table 5-3.

Table 5-3
Underline and overline commands

Function	Control code
Underline ON	<ESC> “_”1
Underline OFF	<ESC> “_”0
Overline ON	<ESC> “_”1
Overline OFF	<ESC> “_”0

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

```

10 ' Demo of underlining and overlining
20 LPRINT "Demo of ";
30 LPRINT CHR$(27);"_-1";
40 LPRINT "UNDELINED";
50 LPRINT CHR$(27);"-0";
60 LPRINT " and ";
70 LPRINT CHR$(27);"_1";
80 LPRINT "OVERLINED";
90 LPRINT CHR$(27);"_0";
100 LPRINT " printing."
```

It should come out like this:

Demo of UNDELINED and OVERLINED printing.

In this program underline is turned on in line 30 with <ESC> “_”1, and then off in line 50 with <ESC> “_”0. The overline is turned on in line 70 with <ESC> “_” 1, and then off in line 90 with <ESC> “_”0. There's a new little wrinkle in this program, though. The semicolons at the end of the 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 overline only part of a line.

■ Boldface printing

Your printer has very good 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 a way to do this

with boldface printing.

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

Table 5-4
Boldface print commands

Function	Control code
Boldface ON	<ESC> "E" or <ESC> "G"
Boldface OFF	<ESC> "F" or <ESC> "H"

Try this short program to see the boldface mode:

```
10 ' Demo of boldface
20 LPRINT CHR$(27);"E";
30 LPRINT "This line is BOLDFACE printing."
40 LPRINT CHR$(27);"F";
50 LPRINT "This line is normal printing."
```

Run this program. The results will look like this:

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

Line 20 turns on boldface with <ESC>"E" and line 30 prints a line of text. In line 40 turns boldface off with <ESC>"F", so your printer is set for normal printing.

Look closely at the different lines of printing. First line, boldface printing, has been printed twice, and they are moved down just slightly the second time they are printed. Now that's pretty nice printing, isn't it?

■ 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 5-5 has the codes for using superscripts and subscripts.

Table 5-5
Superscripts and subscripts commands

Function	Control code
Superscript ON	<ESC>"S"0
Subscript ON	<ESC>"S"1
Super and subscript OFF	<ESC>"T"

Try this program to see them work:

```

10 ' Demo of superscripts and subscripts
20 LPRINT "Look! ";
30 LPRINT CHR$(27);"S0";
40 LPRINT "SUPERSCRIPTS ";
50 LPRINT CHR$(27);"T";
60 LPRINT "& ";
70 LPRINT CHR$(27);"S1";
80 LPRINT "SUBSCRIPTS ";
90 LPRINT CHR$(27);"T";
100 LPRINT "on one line."

```

Look! ^SU_PERSCRIP^TS & SUBSCRIP_TS on one line.

Here line 30 turns on superscripts with <ESC>"S"0. It's turned off in line 50 with <ESC>"T". Then between printing text, subscripts are turned on in line 70 with <ESC>"S"1, and finally off in line 90. 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 136 characters per line.

You can also print 12 characters per inch (*elite pitch*). This gives you 163 characters per line.

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

Table 5-6
Print pitch commands

Pitch	Characters/inch	Control code
Pica	10	<ESC> "P"
Elite	12	<ESC> "M"

Try this program to see how these two pitches work.

```

10 ' Demo of pica and elite pitches
20 LPRINT CHR$(27);"M";
30 LPRINT "This line is ELITE pitch."
40 LPRINT CHR$(27);"P";
50 LPRINT "This line is PICA pitch (normal)."
```

When you run this program you should get this:

```

This line is ELITE pitch.
This line is PICA pitch (normal).
```

Line 20 turns on elite pitch with <ESC> "M". Line 30 prints a line at 12 characters per inch. The <ESC> "P" in line 40 resets the printer to pica pitch and line 50 prints a line in pica pitch.

■ 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 ' Demo of expanded print
20 LPRINT "Demonstration of ";
30 LPRINT CHR$(14);
40 LPRINT "EXPANDED";
50 LPRINT CHR$(20);
60 LPRINT " printing."
70 LPRINT "Notice that ";
80 LPRINT CHR$(14);
90 LPRINT "EXPANDED mode"
100 LPRINT "automatically turns off at the end of a
    line."
```

Demonstration of **EXPANDED** printing. Notice that **EXPANDED mode** automatically turns off at the end of a line.

Expanded print set with CHR\$(14) is automatically cancelled at the end of the line. This is convenient in many applications, such as for one line titles. Note that you didn't need to put an <ESC> in front of the CHR\$(14), although <ESC> CHR\$(14) works just the same.

You can also cancel one line expanded print *before* a carriage return with CHR\$(20), as done in line 50.

Sometimes you may wish to stay in expanded print for more than one line. Change your program to this:

```
10 ' Demo of permanent expanded mode
20 LPRINT CHR$(27);"W1";
30 LPRINT "Permanent expanded"
40 LPRINT "mode stays on until"
50 LPRINT "it is ";
60 LPRINT CHR$(27);"W0";
70 LPRINT "turned off."
```

Now the results look like this:

```
Permanent expanded
mode stays on until
it is turned off.
```

When you turn on expanded print with <ESC> "W"1 it stays on until you turn it off with <ESC> "W"0.

Table 5-7
Expanded print commands

Function	Control code
One line expanded ON	CHR\$(14) or <ESC>CHR\$(14)
One line expanded OFF	CHR\$(20)
Expanded ON	<ESC>"W"1
Expanded OFF	<ESC>"W"0

■ Condensed print

Each of the print pitches also can be condensed to its normal width. This is called condensed print. Try this program to see how it works:

```
10 ' Demo of condensed pitch
20 LPRINT "Demonstration of ";
30 LPRINT CHR$(15);
40 LPRINT "CONDENSED";
50 LPRINT CHR$(18);
60 LPRINT " printing."
```

Demonstration of CONDENSED printing.

Condensed print set with CHR\$(15) stays on until you turn it off with CHR\$(18). Note that you don't need to put an <ESC> in front of the CHR\$(15), although <ESC> CHR\$(15) works just the same.

Table 5-8
Condensed print commands

Function	Control code
Condensed ON	CHR\$(15) <i>or</i> <ESC> CHR\$(15)
Condensed OFF	CHR\$(18)

By combining expanded print and condensed print with the two pitches, this printer has eight different character widths available.

Enter this program to see how the print pitches, expanded print and condensed print can be combined:

```
10 ' Demo of various print pitches
20 LPRINT CHR$(15);
30 LPRINT CHR$(27);"M";
40 LPRINT "This line is CONDENSED ELITE pitch."
50 LPRINT CHR$(27);"P";
60 LPRINT "This line is CONDENSED PICA pitch."
70 LPRINT CHR$(18);
80 LPRINT CHR$(27);"M";
90 LPRINT "This line is NORMAL ELITE pitch."
```

```
100 LPRINT CHR$(27);"P";
110 LPRINT "This line is NORMAL PICA pitch."
120 LPRINT CHR$(27);"W1";
130 LPRINT CHR$(15);
140 LPRINT CHR$(27);"M";
150 LPRINT "This line is EXPANDED CONDENSED ELITE."
160 LPRINT CHR$(27);"P";
170 LPRINT "This line is EXPANDED CONDENSED PICA."
180 LPRINT CHR$(18);
190 LPRINT CHR$(27);"M";
200 LPRINT "This is EXPANDED ELITE."
210 LPRINT CHR$(27);"P";
220 LPRINT "This is EXPANDED PICA."
230 LPRINT CHR$(27);"WO";
```

Here's what you should get from this program:

```
This line is CONDENSED ELITE pitch.
This line is CONDENSED PICA pitch.
This line is NORMAL ELITE pitch.
This line is NORMAL PICA pitch.
This line is EXPANDED CONDENSED ELITE.
This line is EXPANDED CONDENSED PICA.
This is EXPANDED ELITE.
This is EXPANDED PICA.
```

■ Proportional printing

Have you ever noticed in books and magazines? Doesn't it look nice? The main reason is that each character is given an amount of space proportional to its actual width. A typewriter (and most printer), on the other hand, give every character the same amount of space, no matter how wide it is. (Pica pitch, for example, gives a "w" and an "i" 1/10 of an inch each. Look these letters closely and you'll see that a "w" is two or three times as wide as an "i".)

Well, you too enjoy professional-looking proportional printing. You can turn proportional printing on and off with the following command.

Table 5-9
Proportional commands

Function	Control code
Proportional ON	<ESC>"p"1
Proportional OFF	<ESC>"p"0

Try this program to see how the proportional spacing works.

```
10 ' Demo of proportional printing
20 LPRINT CHR$(27);"p1";
30 LPRINT "This line is PROPORTIONAL printing."
40 LPRINT CHR$(27);"p0";
50 LPRINT "This line is NORMAL PICA printing."
```

When you run this program you should get this:

```
This line is PROPORTIONAL printing.
This line is NORMAL PICA printing.
```

In this program line 20 turns on the proportional printing with <ESC>"p"1. Line 30 prints a line with proportional pitch. Then, line 40 resets the proportional printing and line 50 prints a line in normal pica pitch.

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

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> "!" 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 5-10 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 5-10
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	(Not used)	
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:

$$\begin{array}{r}
 \text{Elite} \qquad \qquad 1 \\
 \text{Boldface} \qquad 16 \\
 \text{Expanded} \qquad 32 \\
 \hline
 n = \qquad \qquad 49
 \end{array}$$

The command would look like this:

```
CHR$(27);"!";CHR$(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 Proportional and Condensed modes. Since these cannot combine the printer must make a choice; in this case, the printer chooses Proportional.

Summary notes

- 1) Pica is the default pitch and is active when Elite is turned off.
- 2) When two modes conflict, the one of lesser priority is cancelled. For example, Proportional and Condensed can not be printed at the same time, printing is Proportional.
- 3) Underline, and Expanded modes combine with any print modes.
- 4) Proportional characters are always printed in Boldface.

MEMO

CHAPTER 6

FORMATTING TEXT

Subjects we'll cover in Chapter 6 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, and bottom.

Chapter 5 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 (if DIP switch 2-6 is on; see below).

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.

Now to add a little confusion—most (but not all) versions of BASIC add a line feed (CHR\$(10)) to every carriage return (CHR\$(13)) that they send. If your version of BASIC doesn't do this, then you should turn DIP switch 2-6 off so that your printer will add the line feed for you. When you have DIP switch 2-6 off the printer will do the same thing when it receives a carriage return as it does when it receives a line feed.

If you find that your printer double spaces when it should single space, then you probably need to turn DIP switch 2-6 on.

Table 6-1
Line feed commands

Function	Control code
Return print head to left margin	CHR\$(13)
Advance paper one line	CHR\$(10)

■ Changing the line spacing

When you turn your printer on the line spacing is set to 6 lines per inch (or 8 lines per inch if DIP switch 2-8 is off). 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 ' Demo of line spacing
20 FOR I=1 TO 20
30 IF I=13 THEN 60
40 LPRINT CHR$(27);"A";CHR$(I);CHR$(27);"2";
50 LPRINT "THIS LINE SPACING IS SET TO";I
60 NEXT I
70 LPRINT "THIS LINE SPACING IS SET TO 1/6 INCH."
80 LPRINT CHR$(27);"A";CHR$(10);CHR$(27);"2";
90 END

```

This is what you will get:

```

THIS LINE SPACING IS SET TO 7
THIS LINE SPACING IS SET TO 8
THIS LINE SPACING IS SET TO 9
THIS LINE SPACING IS SET TO 10
THIS LINE SPACING IS SET TO 11
THIS LINE SPACING IS SET TO 12
THIS LINE SPACING IS SET TO 14
THIS LINE SPACING IS SET TO 15
THIS LINE SPACING IS SET TO 16
THIS LINE SPACING IS SET TO 17
THIS LINE SPACING IS SET TO 18
THIS LINE SPACING IS SET TO 19
THIS LINE SPACING IS SET TO 20
THIS LINE SPACING IS SET TO 1/6 INCH.

```

Line 40 defines and changes the line spacing. The command $\langle \text{ESC} \rangle "A" \text{CHR}\(n) defines the line spacing as $n/60$ of an inch; the $\langle \text{ESC} \rangle "2"$ command changes the line spacing to the amount defined by the previous $\langle \text{ESC} \rangle "A"$. The loop that is started in line 20 increases the value of n (the variable I in the program) each time it is executed. So the line spacing increases as the program continues. Line 30 shortcuts the loop when $I = 13$, since BASIC won't let us send $\text{CHR}\$(13)$ without adding an unwanted $\text{CHR}\$(10)$ to it. Finally, the $\langle \text{ESC} \rangle "A" \text{CHR}\(10) and $\langle \text{ESC} \rangle "2"$ commands in line 80 reset the line spacing to 6 lines per inch (10/60 or 1/6 inch).

NOTE: When you look the results closely, the line spacing with the $\langle \text{ESC} \rangle "A" \text{CHR}\(12) command sets the line spacing to 1/6 inch instead of setting 12/60 inch. This is because that the 9-pin printers are normally sets the line spacing to 12/72 inch (1/6 inch) with this command, and most software programs use this command to set the line spacing to 1/6 inch, and we've thought the compatibility with those programs. (The same reason for the $\langle \text{ESC} \rangle "3" \text{CHR}\(36) command.)

Table 6-2
Line spacing commands

Function	Control code
Set line spacing to 1/8 inch	<ESC>"0"
Set line spacing to 7/60 inch	<ESC>"1"
Set line spacing to 1/6 inch or use <ESC>"A" definition	<ESC>"2"
Set or define line spacing to $n/60$ inch	<ESC>"A"CHR\$(n)
Set line spacing to $n/180$ inch	<ESC>"3"CHR\$(n)
One-time line feed of $n/180$ inch	<ESC>"J"CHR\$(n)
One-time reverse line feed of $n/180$ inch	<ESC>"j"CHR\$(n)

■ 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 printhead back to the left margin. The following commands do just that.

The <ESC>"J"CHR\$(n) command causes the printer to make one line feed of $n/180$ inch, but does not *change* the setting of the line spacing. Try this program to see how it works:

```

10 ' Demo of one-time line feeds
20 LPRINT "LINE NUMBER 1."
30 LPRINT "LINE NUMBER 2.";
40 ' One-time line feed
50 LPRINT CHR$(27);"J";CHR$(100);
60 LPRINT "LINE NUMBER 3."
70 LPRINT "LINE NUMBER 4."
80 END

```

Here is what your printer will produce:

```

LINE NUMBER 1.
LINE NUMBER 2.

```

```

LINE NUMBER 3.
LINE NUMBER 4.

```

The `<ESC>"j"CHR$(100)` in line 50 changes the spacing to 100/180 inches for one line only without moving the printhead. The rest of the lines printed with the normal line spacing. Notice that both line 30 and line 50 end with semicolons. This prevents the normal line feed from occurring.

The `<ESC>"j"CHR$(n)` command works the same way except that the paper moves in the opposite direction. Try this simple change to your program and see what a difference it makes!

```
40 ' One-time reverse line feed
50 LPRINT CHR$(27);"j";CHR$(100);
```

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

PAGE CONTROL

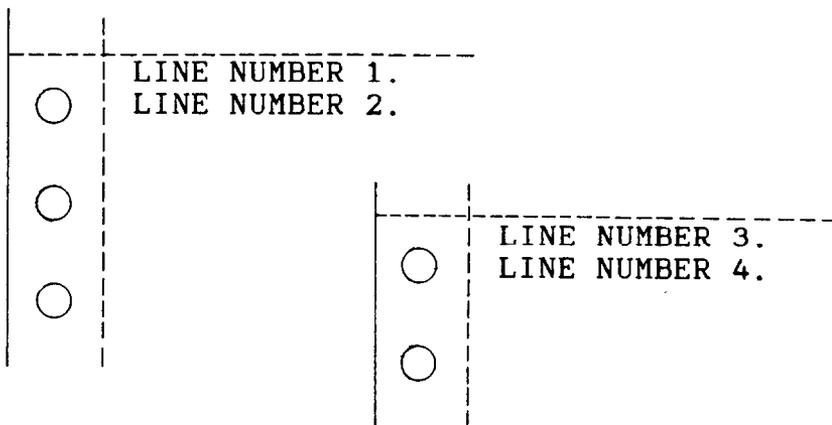
Now that we have seen how to control line spacing, we can go on to page control — positioning the printing on the page and adjusting the paper length.

■ 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 40 and 50 to this:

```
40 ' Form feed
50 LPRINT 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 Chapter 2. When you run the program, the results will look like this:



The form feed (CHR\$(12)) in line 50 caused the printer to move to the top of a new page before printing the last two lines.

■ 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. There are two commands for doing this, shown in this table.

Table 6-3
Form length control

Function	Control code
Advance paper to top of next page	CHR\$(12)
Set the page length to n lines	<ESC>"C"CHR\$(n)
Set the page length to n inches	<ESC>"C"CHR\$(0)CHR\$(n)

Let's set up a 7 inch high form length, which is typical of many computer checks. The following program will do it.

```

10 'Demo of variable form lengths
20 LPRINT CHR$(27);"C";CHR$(0);CHR$(7);
30 LPRINT "PAY TO THE ORDER OF:"
40 LPRINT CHR$(12);
50 LPRINT "PAY TO THE ORDER OF:"
60 END

```

This program should print "PAY TO THE ORDER OF:" twice, and they should be 7 inches apart. Line 20 sets the form length to 7 inches. After line 30 prints, line 40 sends a form feed advance the paper to the top of the next form. Line 50 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).

BOTTOM MARGIN

Many programs that you use a printer don't keep track of where they are printing on the page. This causes a problem when you get to the bottom of a page because these programs just keep on printing, right over the perforation. This makes it very hard to read, especially if a line happens to fall right on the perforation. And if you separate the pages then you are really in trouble.

Of course your printer has a solution to this predicament. This printer can keep track of the position on the page, and advance the paper so that you won't print too near the perforation. There is a command to do this. The control codes to set and cancel are given in the following table.

Table 6-4
Bottom margin commands

Function	Control code
Set bottom margin to n lines	$\langle \text{ESC} \rangle "N" \text{CHR}\(n)
Cancel bottom margin	$\langle \text{ESC} \rangle "O"$

When you set the bottom margin with $\langle \text{ESC} \rangle "N" \text{CHR}\(n) , the value of n tells the printer how many blank lines should be left at the bottom of the page.

Let's try a simple application to see how these margins work. Enter this program, which will print 150 lines *without* bottom margins.

```
10 ' Demo of bottom margin
40 LPRINT CHR$(12);      ' Form feed
```

```
50 FOR I=1 TO 150
60 LPRINT "THIS IS LINE";I
70 NEXT I
90 LPRINT CHR$(12);      ' Form feed
100 END
```

When you run this program it will print 150 lines right down the page and across the perforations. When it's done line 90 sends a form feed to advance to the top of the next page. Look at the lines that have printed near the perforations. Separate the sheets and see if any of the lines have been torn in half. These are the problems that the bottom margin will solve.

Now add the following lines to your program. (Don't forget the semicolons or you won't get quite the same results that we did.)

```
20 ' Leave 6 blank lines at the bottom of page
30 LPRINT CHR$(27);"N";CHR$(6);
80 LPRINT CHR$(27);"O"; ' Clear bottom margin
```

Now when you run the program, your printer skip the last six lines on each page.

Line 30 sets the bottom margin, and line 80 clears bottom margin when we are done.

THIS IS LINE 1
THIS IS LINE 2
THIS IS LINE 3
THIS IS LINE 4
THIS IS LINE 5
THIS IS LINE 6
THIS IS LINE 7
THIS IS LINE 8

THIS IS LINE 52
THIS IS LINE 53
THIS IS LINE 54
THIS IS LINE 55
THIS IS LINE 56
THIS IS LINE 57
THIS IS LINE 58
THIS IS LINE 59
THIS IS LINE 60

THIS IS LINE 61
THIS IS LINE 62
THIS IS LINE 63
THIS IS LINE 64
THIS IS LINE 65
THIS IS LINE 66
THIS IS LINE 67
THIS IS LINE 68

THIS IS LINE 112
THIS IS LINE 113
THIS IS LINE 114
THIS IS LINE 115
THIS IS LINE 116
THIS IS LINE 117
THIS IS LINE 118
THIS IS LINE 119
THIS IS LINE 120

THIS IS LINE 121
THIS IS LINE 122
THIS IS LINE 123
THIS IS LINE 124
THIS IS LINE 125
THIS IS LINE 126
THIS IS LINE 127
THIS IS LINE 128
THIS IS LINE 129
THIS IS LINE 130

SETTING LEFT AND RIGHT MARGINS

The left and right margins of this printer work just like a typewriter — once they are set all the printing is done between them. The commands to set the margins are given in the following table:

Table 6-5
Left and right margin commands

Function	Control code
Set left margin at column n	$\langle \text{ESC} \rangle "1" \text{CHR}\(n)
Set right margin at column n	$\langle \text{ESC} \rangle "Q" \text{CHR}\(n)

Try setting the margins with this program:

```

10 ' Demo of margins
20 GOSUB 70
30 LPRINT CHR$(27);"1";CHR$(10);
40 LPRINT CHR$(27);"Q";CHR$(70);
50 GOSUB 70
60 END
70 FOR I=1 TO 80
80 LPRINT "X";
90 NEXT I
100 LPRINT
110 RETURN

```

The first thing that this program does is to branch to the subroutine that starts in line 70 . 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 30 sets the left margin to 10, and line 40 sets the right margin to 70. 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:

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

When you want to reset the margins to the default values, you have two choices. You can either turn the printer off and back

on, or you can set margin values equal to the default values. This means that you should set a left margin of 0 and right margin of 136 in pica pitch.

If you change the pitch of your printing after you set your margins, the margins will not change. They stay at the same place on the page. So if you set the margins 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 AND VERTICAL 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 both horizontal and vertical tabs which can be used for both text and graphics — and they're really handy for indenting paragraphs and making tables.

■ Horizontal tabs

Horizontal tabs are set automatically every eight positions. To move the print head to the next tab position, send `CHR$(9)`.

Try this program to see how the default tabs work.

```
10 ' Demo of horizontal tabbing
20 LPRINT "ONE";CHR$(9);"TWO";CHR$(9);"THREE";CHR$(9);
   "FOUR"
```

Here's what you should get—

```
ONE      TWO      THREE     FOUR
```

Even though the words are different length, they are spaced out evenly by the horizontal tabs.

Now add the following line to your program to set different horizontal tabs:

```
15 LPRINT CHR$(27); "D"; CHR$(7); CHR$(14); CHR$(21);
   CHR$(0);
```

⟨ESC⟩“D” 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, and 21. The CHR\$(0) 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.

When you run the program now it produces this:

```
ONE      TWO      THREE     FOUR
```

The words are now closer together, but still evenly spaced. To reset the default tabs, use the following program:

```
LPRINT CHR$(27); "e0"CHR$(8)
```

This command sets the horizontal tab to every eight columns.

Table 6-6
Horizontal tab commands

Function	Control code
Advance to next tab position	CHR\$(9)
Set tabs at $n1$, $n2$, etc.	⟨ESC⟩“D”CHR\$($n1$) CHR\$($n2$).....CHR\$(0)
Set tabs every n columns	⟨ESC⟩ “e” 0 CHR\$(n)

■ One-time horizontal tabs

Suppose you need to move to a position across the page, but you only need to do it once. It doesn't make much sense to set up a tab to use only one time. There must be an easier way — and of course there is.

The solution is called a *one-time tab*. Table 6-7 show the three commands.

Table 6-7
One-time horizontal tabs

Function	Control code
One-time horizontal tab of n spaces	$\langle \text{ESC} \rangle$ "f" 0 CHR\$(n)
Absolute horizontal tab	$\langle \text{ESC} \rangle$ "\$" CHR\$($n1$) CHR\$($n2$)
Relative horizontal tab	$\langle \text{ESC} \rangle$ "\ " CHR\$($n1$) CHR\$($n2$)

The first command in Table 6-7 moves the print head n columns to the right. It has the same effect as sending n spaces to the printer.

The absolute horizontal tab command moves the print head to a specified position on the page. The position that you want the print head to move to (measured from the left margin) is specified by the values of $n1$ and $n2$ in 1/60-inch units using the formula $n1 + n2 \times 256$.

Try this program to see how this works.

```
10 ' Demo of absolute horizontal tabs
20 FOR I=60 TO 70 STEP 2
30 LPRINT I;"+++++";
40 LPRINT CHR$(27);"$";CHR$(I);CHR$(0);
50 LPRINT "Horizontal tab."
60 NEXT I
70 END
```

In this program, the print head is positioned before the "H" in "Horizontal" is printed.

```
60 +++++ Horizontal tab.
62 +++++ Horizontal tab.
64 +++++ Horizontal tab.
66 +++++ Horizontal tab.
68 +++++ Horizontal tab.
70 +++++ Horizontal tab.
```

The relative horizontal tab command can move the print head right from the current position. The formula for calculating how much the print head moves is the same as in the absolute horizontal tab command. However, the units by which the print head actually moves vary with the print style you are using. Table 6-8 shows the various options.

Table 6-8
Units of motion for the relative horizontal tab command

Print style	Unit of motion
Draft characters	1/120-inch
LQ characters	1/180-inch
LQ proportional characters	1/360-inch

Try this program to see how this works.

```

10 ' Demo of relative horizontal tabs
20 LPRINT CHR$(27);"x0";
30 LPRINT "Draft characters"
40 GOSUB 120
50 LPRINT CHR$(27);"x1";
60 LPRINT "LQ characters"
70 GOSUB 120
80 LPRINT CHR$(27);"p1";
90 LPRINT "LQ proportional"
100 GOSUB 120
110 END
120 LPRINT "Relative";
130 LPRINT CHR$(27);"\";CHR$(40);CHR$(0);
140 LPRINT "Horizontal";
150 LPRINT CHR$(27);"\";CHR$(20);CHR$(0);
160 LPRINT "Tab."
170 LPRINT
180 RETURN

```

```

Draft characters
Relative Horizontal Tab.

```

```

LQ characters
Relative Horizontal Tab.

```

```

LQ proportional
Relative Horizontal Tab.

```

■ Vertical tabs

Vertical tabs have the same kinds of uses that horizontal tabs do — they just work in the other direction. Horizontal tabs allow you to reach a specific column on the page no matter where you start from. Vertical tabs are the same. If you have a vertical tab

set at line 20, a *vertical tab* (or <VT>) will move you to line 20 whether you start from line 5 or line 19.

The vertical tab is *not* set at the power-on default. If you send a CHR\$(11), which is the ASCII code for <VT>, before we have set up tabs advance the paper one line. Enter this program to see how this works.

```
10 ' Demo of vertical tabs
40 LPRINT CHR$(11);"FIRST TAB."
50 LPRINT CHR$(11);"SECOND TAB."
60 LPRINT CHR$(11);"THIRD TAB."
70 LPRINT CHR$(11);"FOURTH TAB."
```

Now, let's set some vertical tabs of our own. Add these lines to the program:

```
20 LPRINT CHR$(27);"B";CHR$(10);CHR$(15);
30 LPRINT CHR$(25);CHR$(30);CHR$(0);
```

<ESC>"B" is the command to set vertical tabs. Like the horizontal tab setting command, tab positions must be defined in ascending order. Our example sets vertical tabs at lines 10, 15, 25 and 30. Then the CHR\$(11) in each of the following lines advances the paper to the next vertical tab. The printout is shown below.

FIRST TAB.

SECOND TAB.

THIRD TAB.

FOURTH TAB.

Add one more line to the program to demonstrate one more feature of vertical tabs.

```
80 LPRINT CHR$(11);"FIFTH TAB."
```

Now when you run the program the first page looks just like before, but line 80 sends one more <VT> than there are tabs. This doesn't confuse your printer — it advances the paper to the *next* tab position which happens to be the first tab position on the next page. That's nice, isn't it?

There's a one-time vertical tab command that works just like the one-time horizontal tab command. Try this program how these command work.

```
10 ' Demo of print positions
20 LPRINT "* 1st Line, 1st Column"
30 LPRINT CHR$(27);"f1";CHR$(3);
40 LPRINT "* 4th Line, 1st Column"
50 LPRINT CHR$(27);"f0";CHR$(7);
60 LPRINT "* 5th Line, 7th Column"
70 LPRINT CHR$(27);"f1";CHR$(3);
80 LPRINT CHR$(27);"f0";CHR$(5);
90 LPRINT "* 8th Line, 5th Column"
100 END
```

In this program, line 30 sends a one-time vertical tab with <ESC> "f" 1 command. Line 50 sends a one-time horizontal tab with <ESC> "f" 0 command. Finally, line 70 and line 80 send one-time vertical tab and one-time horizontal tab, so the line 90 prints text at the specified position.

```
* 1st Line, 1st Column
```

```
* 4th Line, 1st Column
  * 5th Line, 7th Column
```

```
* 8th Line, 5th Column
```

Table 6-9
Vertical tab commands

Function	Control code
Advance paper to next tab position	CHR\$(11)
Set vertical tabs at $n1$, $n2$, etc.	<ESC>"B"CHR\$($n1$) CHR\$($n2$).....CHR\$(0)
Set vertical tabs every n lines	<ESC> "e" 1 CHR\$(n)
One-time vertical tab	<ESC> "f" 1 CHR\$(n)

■ Vertical tab channels

Vertical tab channels are especially helpful in two situations. The first occurs when you are writing a program to accompany a preprinted form that can accommodate various types of responses. The second occurs when you create a multipage form or report with different vertical tabs on each page.

Table 6-10
Vertical tab channel commands

Function	Control code
Set vertical tabs at $n1$, $n2$, etc. as channel $n0$	<ESC>"b"CHR\$($n0$) CHR\$($n1$) CHR\$($n2$).....CHR\$(0)
Select vertical channel $n0$	<ESC>"f"CHR\$($n0$)

You can store up to eight channels of tab stops. They are numbered from 0 to 7. If you have already stored a set using <ESC>"B" command, your printer has labelled it as channel 0.

Try this program how to see the vertical tab channels work.

```

10 ' Demo of vertical tab channels
20 LPRINT CHR$(27); "b"; CHR$(1); CHR$(10); CHR$(20);
   CHR$(0);
30 LPRINT CHR$(27); "b"; CHR$(2); CHR$(15); CHR$(25);
   CHR$(0);
40 LPRINT CHR$(27); "b"; CHR$(3); CHR$(17); CHR$(28);
   CHR$(0);
50 ' Use vertical tab channels
60 FOR I=1 TO 3
70 LPRINT "TOP OF FORM"
80 LPRINT CHR$(27); "/" ; CHR$(I);
90 LPRINT CHR$(11);
100 LPRINT "1ST TAB OF CHANNEL"; I
110 LPRINT CHR$(11);

```

```
120 LPRINT "2ND TAB OF CHANNEL";I
130 LPRINT CHR$(12);
140 NEXT I
150 LPRINT CHR$(27);"@"
160 END
```

When you run this program you should get like this.

TOP OF FORM	TOP OF FORM	TOP OF FORM
1ST TAB OF CHANNEL 1		
	1ST TAB OF CHANNEL 2	
2ND TAB OF CHANNEL 1		1ST TAB OF CHANNEL 3
	2ND TAB OF CHANNEL 2	
		2ND TAB OF CHANNEL 3

In this program we set tabs at 10 and 20 in channel 1 in line 20. In line 30 we set tabs 15 and 25 in channel 2, and in line 40 we set tabs 17 and 28 in channel 3.

Because the channels are stored, you must make the printer to recall one before you use it, so we used `<ESC>"P"CHR$(n0)` in line 80.

MEMO

CHAPTER 7

SPECIAL FEATURES OF THE PRINTER

Subjects we'll cover in Chapter 7 include—

- **Printer's bell;**
- **Master reset;**
- **Uni-directional printing;**
- **International character sets;**
- **Printing BIG characters;**
- **The optional sheet feeder;**
- **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 re-

mind an operator to change the paper or to make another adjustment to the printer.

You can try this by typing:

```
LPRINT 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> "@"` will reset all of the printer's features to the power on condition (as determined by the DIP switches), with one exception. This exception is that `<ESC> "@"` will not erase any characters that you have stored in the printer's RAM memory.

In addition, if you set the "Panel" modes by the control panel settings at the power on, these functions will be remain with this control code.

■ Putting your printer to sleep

You know how to put your printer *off line* with the On Line key on the control panel. Your printer has another *off line* state that can be controlled from your computer. When you turn the printer *off line* from your computer, this printer will ignore anything that you send it, except for the code to go *on line* again. `CHR$(19)` is the code to turn your printer off line; `CHR$(17)` turns your printer to on line status.

■ 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 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 7-1
Some miscellaneous commands

Function	Control code
Sound bell	CHR\$(7)
Master reset	<ESC> "@"
Off line	CHR\$(19)
On line	CHR\$(17)
Paper-out detector off	<ESC> "8"
Paper-out detector on	<ESC> "9"
Move print head back one space	CHR\$(8)
Delete last character sent	CHR\$(127)
Cancel text in print buffer	CHR\$(24)
Immediate-print on	<ESC> "i"1
Immediate-print off	<ESC> "i"0
Add <i>n</i> dot spaces between characters	<ESC> CHR\$(32) CHR\$(<i>n</i>)
Repeat characters	<ESC> "V" CHR\$(<i>n</i>) CHR\$(<i>m1</i>) ... CHR\$(<i>mx</i>) <ESC> "V" CHR\$(0)

■ Backspace, delete, and cancel text

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

Delete (CHR\$(127)) also "backs up" one character, but then it "erases" the previous character (it's erased from your printer's buffer, not from the paper).

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 works.

```

10 ' Backspace, delete, and cancel line
20 LPRINT "BACKSPACE DOES NOT";
30 LPRINT CHR$(8);CHR$(8);CHR$(8);
40 LPRINT "=== WORK"
50 LPRINT "DELETE DOES NOT";
60 LPRINT CHR$(127);CHR$(127);CHR$(127);
70 LPRINT "WORK"
80 LPRINT "CANCEL LINE";

```

```
90 LPRINT CHR$(24);
100 LPRINT "DOES NOT WORK"
```

Here is what this program will print:

```
BACKSPACE DOES NOT WORK
DELETE DOES WORK
DOES NOT WORK
```

The backspace codes in line 30 move the printhead a total of three spaces to the left so that the first part of line 40 will overprint the word "NOT". The delete codes in line 60 "erase" the three letters in the word "NOT" so that it doesn't even print.

In line 90, CHR\$(24) deletes the words in line 80. The semicolon at the end of line 80 prevents a line feed from causing that line to print before the printer receives the CHR\$(24) code. The text in line 100 prints as it normally would because it is after CHR\$(24).

■ Immediate-print

This printer can print at fine rate of 250 characters per second. But it will also print more slowly at the speed of your typing. In *immediate-print* mode, the printhead prints one character at a time, as you send it. This printer also moves the paper up so that you can see the current line and then down to continue printing.

You can turn immediate-print mode on with <ESC> "i" 1. But before looking at it, let's review the normal operation of the print buffer. Enter this program.

```
20 A$="" :INPUT " TYPE A CHARACTER ",A$
30 IF A$="" THEN 50
40 LPRINT A$; : GOTO 20
50 LPRINT : LPRINT CHR$(27);"@"
```

Now type several characters, and after each press the RETURN key. True to form, the printer just stuffs the characters into its buffer while it waits for a carriage return code. (In this program the RETURN key doesn't send a car-

riage return code.) To end this program and print the contents of the buffer, press RETURN alone.

Now add this line:

```
10 LPRINT CHR$(27);"i1";
```

And RUN the program. Your printer responds to your typing — immediately.

When you are finished, press RETURN alone.

■ Adjusting the width of space between characters

This printer provides a command that adjusts the space between all the characters that it prints when the DIP switch 2-2 is set off. The `<ESC> CHR$(32) CHR$(n)` command adds blank space between the characters. The units of space that are added vary depending on the current print style, as shown in Table 7-2.

Table 7-2
Unit of adjusting space width

Print style	Unit of motion
Draft characters	1/120-inch
LQ characters	1/180-inch
LQ proportional characters	1/360-inch

Try this program to see how this works.

```
10 ' Demo of adjusting spaces
20 FOR I=5 TO 1 STEP -1
30 LPRINT CHR$(27);" ";CHR$(I);
40 LPRINT "This line is added";I;"spaces."
50 NEXT I
60 LPRINT CHR$(27);" ";CHR$(0);
70 LPRINT "This line is Normal space."
80 END
```

```
This line is added 5 spaces.
This line is added 4 spaces.
This line is added 3 spaces.
This line is added 2 spaces.
This line is added 1 spaces.
This line is Normal space.
```

This command can be used to produce micro-justification, which is a method of justifying lines by increasing the space between each character.

■ Repeat characters

This printer has a command that automatically repeats characters for you. Try this program to see how it works.

```
10 ' Demo of character repeat
20 LPRINT CHR$(27);"V";CHR$(4);
30 LPRINT "This line shows Repeat characters."
40 LPRINT CHR$(27);"V";CHR$(0);
50 LPRINT "And this is not."
60 END
```

In this program, line 20 selects the 4 times repeat with $\langle \text{ESC} \rangle$ "V" CHR\$(4) command. And line 30 sets the characters for repeating. Then line 40 terminates this command with $\langle \text{ESC} \rangle$ "V" CHR\$(0). The results from this program look like this.

```
This line shows Repeat characters.
And this is not.
```

■ Uni-directional printing

Uni-directional printing is a 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 7-3
Printing direction commands

Function	Control code
Print in one direction	$\langle \text{ESC} \rangle$ "U"1
Print in both directions	$\langle \text{ESC} \rangle$ "U"0
One time print in one direction	$\langle \text{ESC} \rangle$ "<"

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

```
10 ' Uni-directional printing
20 LPRINT CHR$(27);"A";CHR$(6);
30 LPRINT CHR$(27);"2";
40 FOR I=1 TO 10
50 LPRINT ":"
60 NEXT I
70 LPRINT :LPRINT
80 LPRINT CHR$(27);"U1";
90 FOR I=1 TO 10
100 LPRINT ":"
110 NEXT I
120 LPRINT CHR$(12);CHR$(27);"@"
```

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 20 defines the line spacing to 6/60 of an inch so that the characters that we print will touch top to bottom, and line 30 activates the definition. 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.

You can also set the printer to print in one direction for one line only by using the `<ESC> “<”` command. This command immediately moves the printhead to the left margin and then prints the remainder of the line from left to right.

■ The seven bit dilemma

Certain computers (but not the IBM-PC fortunately!) don't have the capability to send eight bits on their parallel interface. They can only send seven bits. This would make it impossible for these computers to use this printer's block graphics characters and special symbols if our engineers hadn't thought of a solution. (All of these characters have ASCII codes greater than 127 which means that the eighth bit must be on to use them.) The solution lies in the three control codes given in the following table:

Table 7-4
Eighth bit controls

Function	Control code
Turn the eighth bit ON	<code><ESC> “>”</code>
Turn the eighth bit OFF	<code><ESC> “=”</code>
Accept the eighth bit “as is” from the computer	<code><ESC> “#”</code>

■ Block graphics characters and special symbols

Besides the upper and lower case letters and symbols that we are by now familiar with, your printer has a whole different set of characters that are for special uses. These characters include block graphics for drawing forms and graphs, and special symbols for mathematical, engineering and professional uses. The special characters are included in two character sets. The character set you normally use is called character set #1. The special characters are printed out when you send ASCII codes 160~255 to the printer.

Your printer also offers character set #2 which is almost the same as character set #1 except for the addition of ASCII codes 3~6, 21, and 128~159. Character set #2 is selected with `<ESC> “6”`; to go back to character set #1, use `<ESC> “7”`.

You can also specify the power-on default character set by setting DIP switch 1-5 on for character set #1 and off for character set #2 while the DIP switch 2-2 is set on. The following program will print out all of the graphics characters available:

```

10 LPRINT CHR$(27);"0";
20 LPRINT CHR$(27);"6";
30 FOR J=3 TO 6
40 LPRINT " ";J;CHR$(J);CHR$(9);
50 NEXT J
60 LPRINT " 21 ";CHR$(21)
70 LPRINT
80 FOR J=128 TO 254 STEP 5
90 FOR I=J TO J+4
95 IF I>254 THEN 110
100 LPRINT I;CHR$(I);CHR$(9);
110 NEXT I
120 LPRINT :LPRINT
130 NEXT J

```

Figure 7-1 shows what this program will print. If your chart doesn't look like this because it has regular letters and numbers instead of the special symbols, then your computer is only using seven bits (unless you have set DIP switches 2-2 and 2-5 off by mistake). You can get the correct printout by changing line 100 to this:

```

100 LPRINT I;CHR$(27);">";CHR$(I);CHR$(27);"=";CHR$(9);

```

So how are all of these strange characters used? Here is a short program that demonstrate how the graphics characters can be combined to create a figure: the 5 of clubs.

```

10 LPRINT CHR$(27);"6";
20 LPRINT CHR$(218);CHR$(196);CHR$(196);CHR$(196);
   CHR$(196);CHR$(196);CHR$(191)
30 LPRINT CHR$(179);CHR$(53);CHR$(32);CHR$(32);
   CHR$(32);CHR$(32);CHR$(179)
40 LPRINT CHR$(179);CHR$(32);CHR$(5);CHR$(32);CHR$(5);
   CHR$(32);CHR$(179)
50 LPRINT CHR$(179);CHR$(32);CHR$(32);CHR$(5);
   CHR$(32);CHR$(32);CHR$(179)
60 LPRINT CHR$(179);CHR$(32);CHR$(5);CHR$(32);CHR$(5);
   CHR$(32);CHR$(179)
70 LPRINT CHR$(179);CHR$(32);CHR$(32);CHR$(32);
   CHR$(32);CHR$(53);CHR$(179)
80 LPRINT CHR$(192);CHR$(196);CHR$(196);CHR$(196);
   CHR$(196);CHR$(196);CHR$(217)
90 LPRINT CHR$(27);"7"

```

3 ♡	4 ♦	5 ♣	6 ♠	21 §
128 Ç	129 ü	130 é	131 â	132 ä
133 à	134 å	135 ç	136 ê	137 ë
138 è	139 ï	140 î	141 ì	142 Å
143 A	144 E	145 æ	146 Æ	147 ô
148 ö	149 ò	150 û	151 ù	152 ÿ
153 Ö	154 Ü	155 ç	156 £	157 ¥
158 R̄	159 f	160 á	161 í	162 ó
163 ú	164 ñ	165 Ñ	166 a	167 Ω
168 ž	169 ˘	170 ˘	171 ½	172 ¼
173 i	174 «	175 »	176	177 █
178 █	179	180 †	181 †	182 †
183 †	184 †	185 †	186	187 †
188 †	189 †	190 †	191 †	192 †
193 †	194 †	195 †	196 †	197 †
198 †	199 †	200 †	201 †	202 †
203 †	204 †	205 =	206 †	207 †
208 †	209 †	210 †	211 †	212 †
213 †	214 †	215 †	216 †	217 †
218 †	219 █	220 █	221 █	222 █
223 █	224 α	225 β	226 Γ	227 π
228 Σ	229 σ	230 μ	231 τ	232 Φ
233 θ	234 Ω	235 δ	236 ∞	237 ø
238 €	239 ∩	240 ≡	241 ±	242 ≥
243 ≤	244 ∫	245 J	246 ÷	247 ≈
248 °	249 ·	250 -	251 √	252 °
253 ²	254 ■			

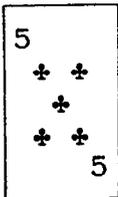
Figure 7-1. Character set #2

If you have a 7-bit interface, add the following lines to the program given above:

```
15 LPRINT CHR$(27);">";
95 LPRINT CHR$(27);"="
```

In this program line 10 selects character set #2 so that all the graphics characters can be used (including the "club" symbol). Then lines 20~80 print the 5 of clubs figure. Line 90 cancels character set #2 (which is the same as selecting character set #1).

Here is what this program prints:



■ International character sets

Your printer is a multi-lingual printer for it can speak in eleven languages! Your printer changes languages by changing 12 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 7-5.

Table 7-5
International character set commands

Country	Control code
U.S.A	<ESC> "R"CHR\$(0)
France	<ESC> "R"CHR\$(1)
Germany	<ESC> "R"CHR\$(2)
England	<ESC> "R"CHR\$(3)
Denmark type I	<ESC> "R"CHR\$(4)
Sweden	<ESC> "R"CHR\$(5)
Italy	<ESC> "R"CHR\$(6)
Spain	<ESC> "R"CHR\$(7)
Japan	<ESC> "R"CHR\$(8)
Norway	<ESC> "R"CHR\$(9)
Denmark type II	<ESC> "R"CHR\$(10)

The characters that change are shown beneath their ASCII code in Table 7-6.

Table 7-6
International character sets

Country	35	36	64	91	92	93	94	96	123	124	125	126
U.S.A	#	\$	@	[\]	^	`	{		}	~
France	#	\$	à	·	ç	§	^	`	é	ù	è	..
Germany	#	\$	§	Ä	ö	ü	^	`	ä	ö	ü	ß
England	£	\$	@	[\]	^	`	{		}	~
Denmark type I	#	\$	@	Æ	Ø	Å	^	`	æ	ø	å	~
Sweden	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Italy	#	\$	@	·	\	é	^	ù	à	ò	è	ì
Spain	₧	\$	@	ı	Ñ	¿	^	`	ñ	}	~	
Japan	#	\$	@	[¥]	^	`	{		}	~
Norway	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
Denmark type II	#	\$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü

■ 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 7-7
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 ' BIG character
20 LPRINT: LPRINT
30 LPRINT CHR$(27);"-1";
40 LPRINT "THIS IS ";
50 LPRINT CHR$(27);"h";CHR$(1);
60 LPRINT "DOUBLE";
70 LPRINT CHR$(27);"h";CHR$(0);
80 LPRINT " SIZED PRINTING."
```

```

90 LPRINT "THIS IS ";
100 LPRINT CHR$(27);"h";CHR$(2);
110 LPRINT "QUAD";
120 LPRINT CHR$(27);"h";CHR$(0);
130 LPRINT " SIZED PRINTING."
140 LPRINT CHR$(27);"-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 ' Demo of aliging big characters
20 LPRINT CHR$(27);"-1";
30 LPRINT "THIS IS ";
40 LPRINT CHR$(27);"j";CHR$(24);
50 LPRINT CHR$(27);"h";CHR$(1);
60 LPRINT "DOUBLE";
70 LPRINT CHR$(27);"h";CHR$(0);
80 LPRINT " SIZED PRINTING."
90 LPRINT :LPRINT :LPRINT
100 LPRINT "THIS IS ";
110 LPRINT CHR$(27);"j";CHR$(72);
120 LPRINT CHR$(27);"h";CHR$(2);
130 LPRINT "QUAD";
140 LPRINT CHR$(27);"h";CHR$(0);
150 LPRINT " SIZED PRINTING."
160 LPRINT CHR$(27);"-0"

```

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

THIS IS **DOUBLE** SIZED PRINTING.

THIS IS **QUAD** SIZED PRINTING.

■ 9-pin graphics emulation mode

As you had learned in Chapter 2, this printer has a *unique* 9-pin graphics emulation mode.

You can select this mode with a control code — $\langle \text{ESC} \rangle$ “g”. After you’ve entered into the 9-pin graphics emulation mode, following commands work in the same manner for the 9-pin printers, as shown below:

- $\langle \text{ESC} \rangle$ “A” *n* — Sets or defines line spacing to *n*/72 inch.
- $\langle \text{ESC} \rangle$ “3” *n* — Sets line spacing to *n*/216 inch.
- $\langle \text{ESC} \rangle$ “J” *n* — Sends a one-time paper feed of *n*/216 inch.
- $\langle \text{ESC} \rangle$ “j” *n* — Sends a one-time reverse feed of *n*/216 inch.

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

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

■ 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 7-8
Automatic sheet feeder commands

Function	Control code
Select automatic feed mode	$\langle \text{ESC} \rangle$ CHR\$(25) CHR\$(4) or “((4))”
Cancel automatic feed mode	$\langle \text{ESC} \rangle$ CHR\$(25) CHR\$(0) or “((0))”
Insert paper from first bin	$\langle \text{ESC} \rangle$ CHR\$(25) CHR\$(1) or “((1))”
Insert paper from second bin	$\langle \text{ESC} \rangle$ CHR\$(25) CHR\$(2) or “((2))”
Eject paper	$\langle \text{ESC} \rangle$ CHR\$(25)“R” or “((R))”

When the automatic sheet feeder is installed, you must set the DIP switch 2-4 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
- Top and bottom margins
- Vertical tab settings

Caution: The five-character commands (e.g. “((0)”) work as the automatic sheet feeder control codes whenever the automatic sheet feeder is mounted on the printer (e.g. LLISTing).

On the other hand, the automatic sheet feeder is not mounted on the printer, those five characters are printed as regular characters.

■ Reading a hex dump

We’ve seen how to make a hex dump in Chapter 2, 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 and run the following program. Hold down both the Paper Feed key and Mode key and turn on the printer.

```
10 FOR I=0 TO 255
20 LPRINT CHR$(I)
30 NEXT I
40 LPRINT
50 END
```

If your system passes the codes directly to the printer without changing them, you will get like this. (You can print out the last remaining line in the print buffer by putting the printer off line with the On Line key.)

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
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
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	AB	AC	AD	AE	AF			AA	AB	AC	AD	AE	AF		
BA	BB	BC	BD	BE	BF			BB	BC	BD	BE	BF			
CA	CB	CC	CD	CE	CF			CB	CC	CD	CE	CF			
DA	DB	DC	DD	DE	DF			DB	DC	DD	DE	DF			
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
00	0A														

Most BASICs, however, are not quite that straight forward. For example, the IBM-PC prints like the following.

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
10	11	12	13	14	15	16	17	17	18	19	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
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
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	AB	AC	AD	AE	AF			AA	AB	AC	AD	AE	AF		
BA	BB	BC	BD	BE	BF			BB	BC	BD	BE	BF			
CA	CB	CC	CD	CE	CF			CB	CC	CD	CE	CF			
DA	DB	DC	DD	DE	DF			DB	DC	DD	DE	DF			
E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF
00	0A														

Notice that your printer is receiving decimal code 13 (hex 0D) is coming with hex 0A, which is really decimal 10. In addition, your printer does not receive decimal code 26 (hex 1A).

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 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.

CHAPTER 8

CREATING YOUR OWN CHARACTERS

Subjects we'll cover in Chapter 8 include—

- **Designing your own characters;**
- **Printing your own characters;**
- **Print mode and the download characters.**

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 7, 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. The process is called dot-matrix printing because each character is composed of small dots arranged on a matrix, or grid. To see how that character grid gets printed, you need to take a look at the print head itself.

The printhead in this printer contains 24 wires apparently arranged in a vertical column. If you were to remove the print head and look at the wires, you would see two staggered columns of twelve wires. However, the timing is such that they print as one continuous column of 24 wires.

Figure 8-1 shows an enlarged schematic view of the front of the printhead, showing the ends of the wires and their relationship to the draft characters. As you can see, the capital letters use the top eighteen wires of the printhead, and the descenders (such as the lower case “p” shown) use the bottom eighteen pins. As the printhead 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 printhead causes the appropriate wire to strike the ribbon (making this printer an impact printer).

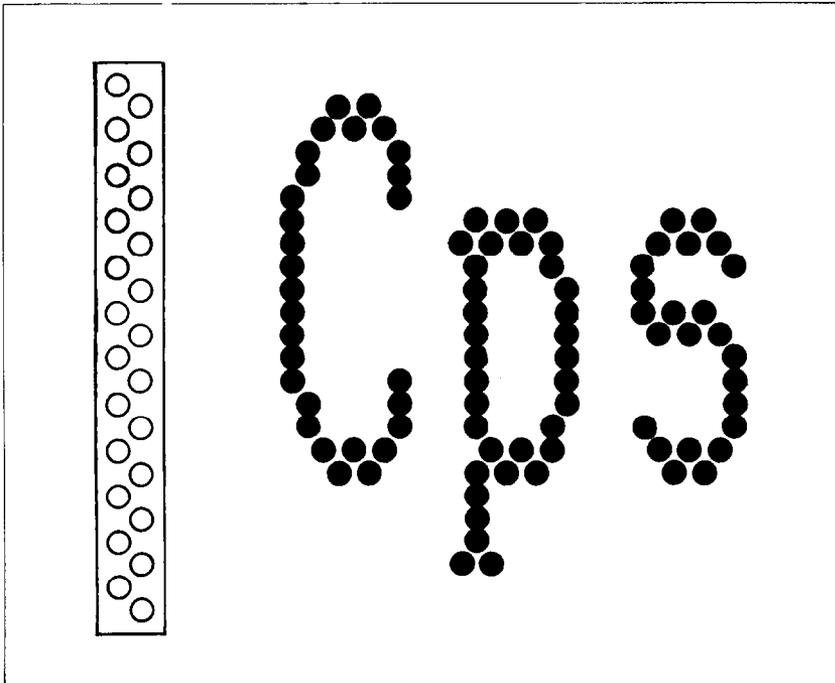


Figure 8-1. As the printhead moves across the page, each of the wires prints one row of dots.

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 LQ 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 draft characters, whether it is from the standard ROM or in download RAM, is constructed on a grid which is five “boxes” by twenty-four “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 8-2. 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 9 dots wide by 24 dots high.

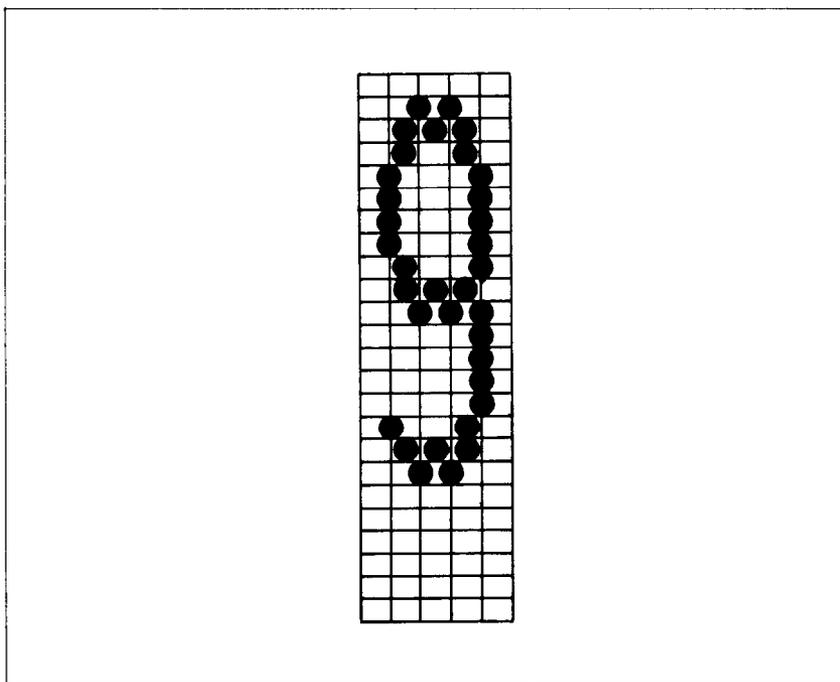


Figure 8-2. Dots can be inside boxes or straddle the vertical lines of the grid.

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 8-3 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 an automobile as an example of a draft download character. You can see how we've laid it out in Figure 8-4. Later in this chapter we'll use this character to create a small graph.)

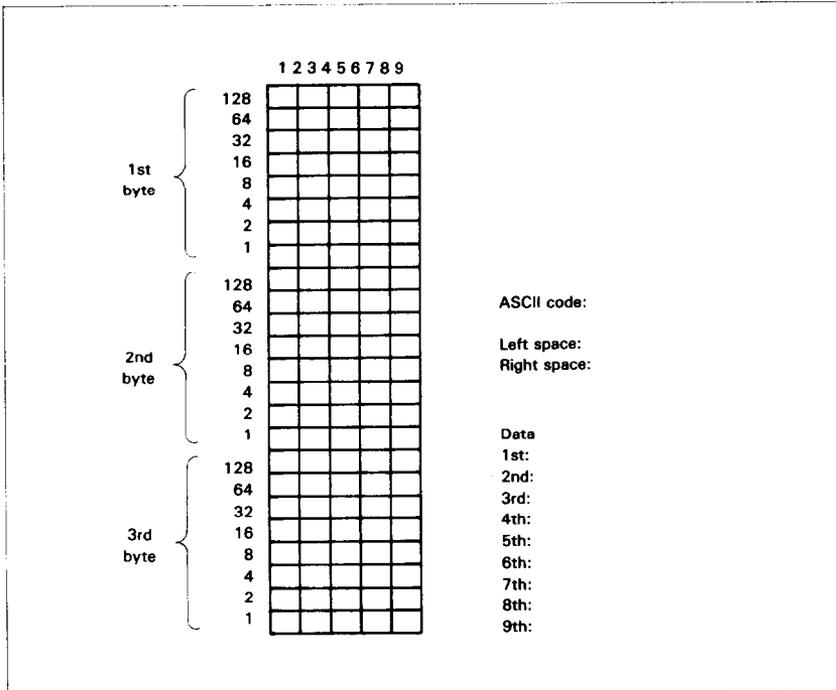


Figure 8-3. Use this grid (or one similar to it) to define your own draft characters.

You'll notice that Figure 8-3 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.

■ Rule 1: Dots cannot overlap

As you can see in Figure 8-4 our car 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 8-5? Because the dots that straddle the vertical lines in the grid actually overlap those inside the boxes. If we tried to print overlapping

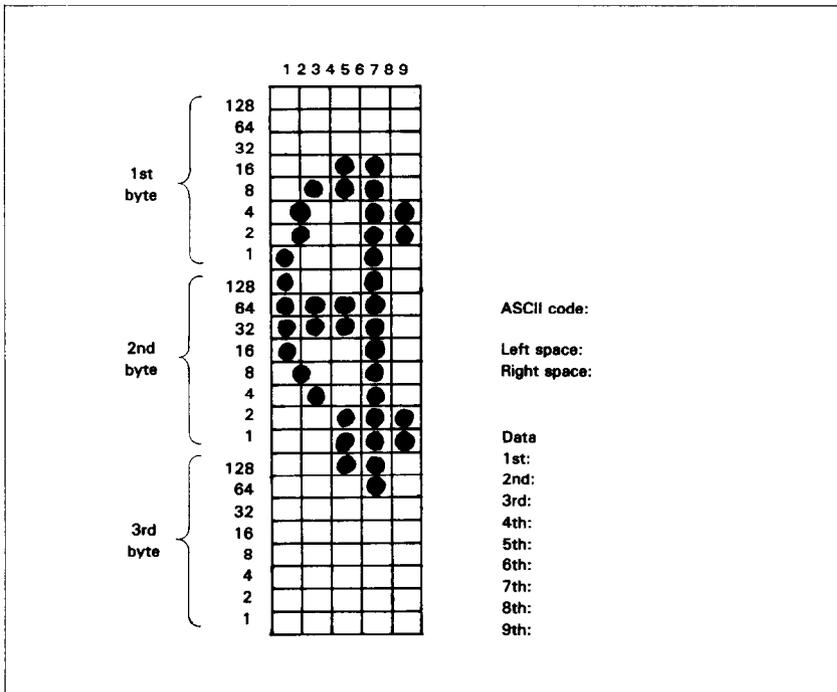


Figure 8-4. We've designed an automobile character.

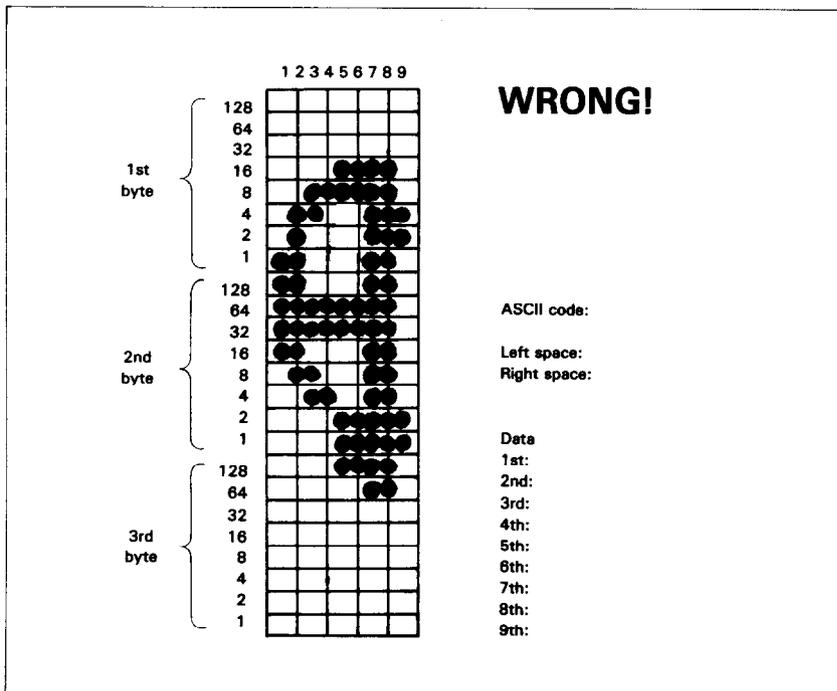


Figure 8-5. Dots cannot overlap; those in immediately adjacent "half columns" will be ignored when the character is printed.

dots, the printhead 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 8-5. (Actually, you can define it, but when it prints, your printer will leave out the overlapping dots, so that it would print like Figure 8-4.)

■ **Rule 2: Each row is divided into three bytes**

Now it's time to give our creative side a break and get down to some basic arithmetic. Each vertical column (which has a maximum of 24 dots) is first divided into three groups of eight dots. Each group of eight dots is represented by one byte, which consists of eight bits. 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 8-6, 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.

128			● — 128
64	● — 64		● — 64
32	● — 32	● — 32	● — 32
16		● — 16	● — 16
8		● — 8	● — 8
4	● — 4		● — 4
2	● — 2	● — 2	● — 2
1	● — 1		● — 1
Sum	<hr/> 103	<hr/> 58	<hr/> 255

Figure 8-6. 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 8-7 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: *d1*, *d2*, *d3*, etc. These labels correspond to the labels in the command syntax statement,

which we'll get to shortly.

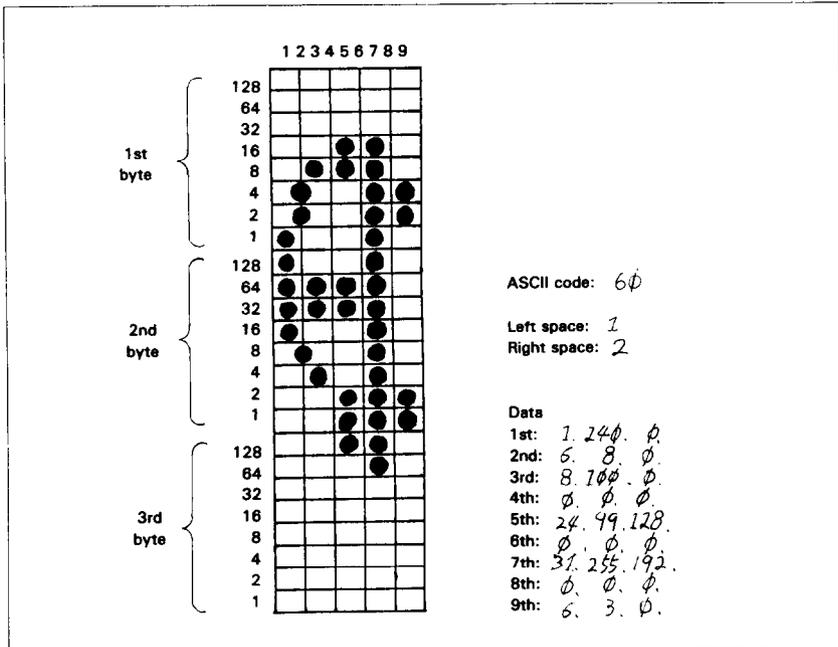


Figure 8-7. Add the values of the dots in each column and write the sum of each column at the bottom.

■ Assigning a value of character space

Besides being able to specify the actual width of the character, this printer allows you to specify the position in the standard grid where the character will print. You must specify the dot column in which the printed character starts and the dot column in which the character ends. Why, you may ask, would you want to define a character this way instead of merely defining the overall width of the character? Because this printer's draft proportional character definitions can also be used to print normal width characters, which are twelve dot columns wide. And by centering even the narrow characters in the complete grid they will look good even you aren't printing them proportionally.

The three bytes are used to specify the width of the character and the space to be allowed on either side of it. The left space (in dot columns) is specified by *m0* and the right space is specified by *m2*. The second byte (*m1*) specifies the number of columns of dots that will be printed by the character. By varying the width of the character itself and the spaces around it, you can actually

create proportional width characters that print at draft speed.

When defining draft characters, the number of printed columns ($m1$) cannot exceed 9, and the sum of $m0 + m1 + m2$ can not exceed 12.

■ 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 128 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 code — numbers from 0 to 255. For the download character sets you can define any positions between 0 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.

The entire range of ASCII codes (from 0 to 127) can be used

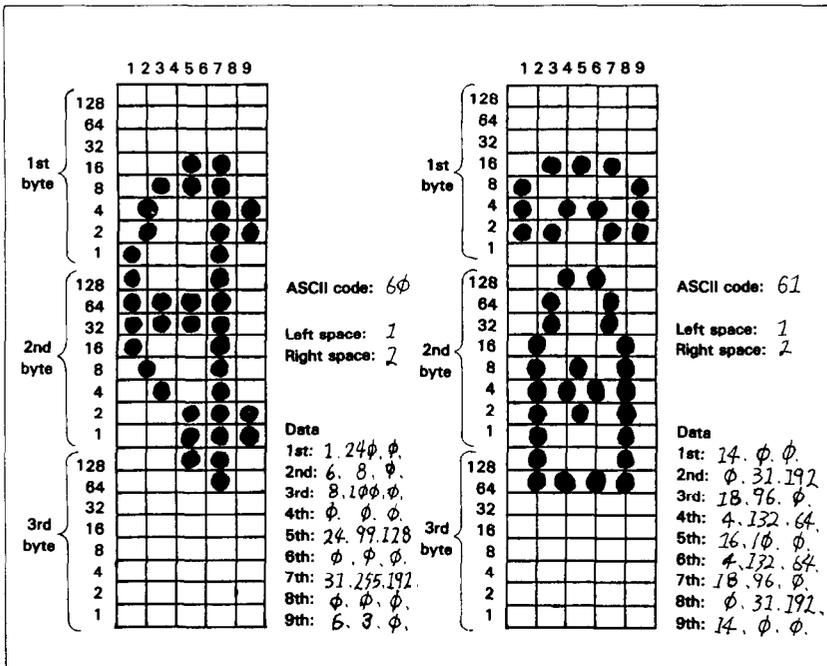


Figure 8-8. Character designs for the two graph symbols.

for user-defined characters, but a caution is in order. Characters with ASCII values below 32 are usually reserved as control codes by both your computer and your printer. While these characters can be replaced by user-defined characters, it should be avoided because of the effect on the normal control codes.

In our example, we'll assign the car 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 car, so in Figure 8-8 we've made completed grids for some other symbols: a telephone (quite a strange mix of characters!). The information on the grids is now completed.

■ 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 d1 d2 d3 ... dx
```

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.

The three bytes following the specification of the range of characters (*m0*, *m1* and *m2*) are used to specify the width of the character and the space to be allowed on either side of it. The left space (in dot column) is specified by *m0* and the right space is specified by *m2*. The second byte (*m1*) specifies the number of columns of dots that will be printed by the character. By varying the width of the character itself and the spaces around it, you can actually create proportional width characters.

You'll probably recognize *d1 d2 ... dx* from the top of our layout grid. That's right, each column is described by three bytes. Now we've got everything we need to download one character to the printer. The complete command for our car character is shown below:

```
CHR$(27);CHR$(38);CHR$(0);CHR$(60);CHR$(60);
CHR$(1);CHR$(9);CHR$(2);CHR$(1);CHR$(240);CHR$(0);
CHR$(6);CHR$(8);CHR$(0);CHR$(8);CHR$(100);CHR$(0);
CHR$(0);CHR$(0);CHR$(0);CHR$(24);CHR$(99);CHR$(128);
CHR$(0);CHR$(0);CHR$(0);CHR$(31);CHR$(255);CHR$(192);
CHR$(0);CHR$(0);CHR$(0);CHR$(6);CHR$(3);CHR$(0)
```

Now let's send the information to the printer. The following program will send the character definitions for all two characters to the printer. Before you enter the program, turn off the printer and set the DIP switch 2-3 off. Otherwise, your printer will ignore your download definitions.

After you set the DIP switch correctly, enter the program and run it.

```
10 LPRINT CHR$(27);"x0";
20 LPRINT CHR$(27);"&";CHR$(0);CHR$(60);CHR$(61);
30 FOR N=60 TO 61
40 FOR M=1 TO 30
50 READ MM
60 LPRINT CHR$(MM);
70 NEXT M
80 NEXT N
90 LPRINT
100 END
110 DATA 1, 9, 2, 1,240, 0, 6, 8, 0, 8,
100, 0
120 DATA 0, 0, 0, 24, 99,128, 0, 0, 0, 31,
255,192
130 DATA 0, 0, 0, 6, 3, 0
140 DATA 1, 9, 2, 14, 0, 0, 0, 31,192, 18,
96, 0
150 DATA 4,132, 64, 16, 10, 0, 4,132, 64, 18,
96, 0
160 DATA 0, 31,192, 14, 0, 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 two characters to your printer. But how do you know that? If you try printing those characters now you don't get a car and telephone. 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 LPRINT CHR$(27);"1";CHR$(0);
20 LPRINT CHR$(60);CHR$(61);
30 LPRINT CHR$(27);"0";CHR$(0);
40 END
```

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

```

```

(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 LPRINT CHR$(27);"1";CHR$(0)
20 FOR I=32 TO 126
30 LPRINT CHR$(I);
40 NEXT I
```

```

50 LPRINT
60 FOR I=160 TO 254
70 LPRINT CHR$(I);
80 NEXT I
90 LPRINT
100 LPRINT CHR$(27);";%0";CHR$(0)
110 END

```

Nope! Just two 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:

```
5 LPRINT CHR$(27);":":CHR$(0);CHR$(0);CHR$(0)
```

Now try the download printout test program again. Your results look like Figure 8-9.

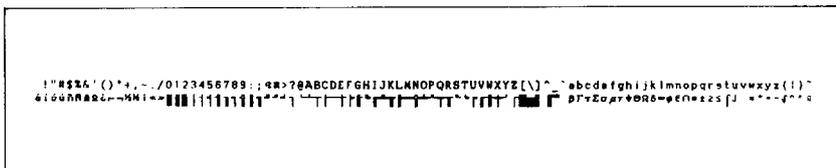


Figure 8-9. Printout of the download character set, into which all the ASCII characters have been copied, and the < and = have been changed.

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 LPRINT CHR$(27);"x0";
20 LPRINT CHR$(27);":":CHR$(0);CHR$(0);CHR$(0)
30 LPRINT CHR$(27);"&";CHR$(0);CHR$(60);CHR$(61);
40 FOR N=60 TO 61
50 FOR M=1 TO 30
60 READ MM
70 LPRINT CHR$(MM);
80 NEXT M
90 NEXT N
100 LPRINT
110 DATA 1, 9, 2, 1,240, 0, 6, 8, 0, 8,
100, 0
120 DATA 0, 0, 0, 24, 99,128, 0, 0, 0, 31,
255,192
130 DATA 0, 0, 0, 6, 3, 0
140 DATA 1, 9, 2, 14, 0, 0, 0, 31,192, 18,
96, 0
150 DATA 4,132, 64, 16, 10, 0, 4,132, 64, 18,
96, 0
160 DATA 0, 31,192, 14, 0, 0
170 LPRINT CHR$(27);"D";CHR$(11);CHR$(0)
180 LPRINT CHR$(27);"h";CHR$(1);
190 LPRINT " DIFFUSION RANGES OF"
200 LPRINT " CARS & TELEPHONES"
210 LPRINT CHR$(27);"h";CHR$(0);
220 LPRINT CHR$(27);"%1";CHR$(0);
230 LPRINT "USA";CHR$(9);
240 FOR I=0 TO 681 STEP 50
250 LPRINT CHR$(60);
260 NEXT I
270 LPRINT
280 LPRINT CHR$(9);
290 FOR I=0 TO 781 STEP 50
300 LPRINT CHR$(61);
310 NEXT I
320 LPRINT
330 LPRINT "GERMANY";CHR$(9);
340 FOR I=0 TO 412 STEP 50
350 LPRINT CHR$(60);
360 NEXT I
```


Note that we didn't *have* to re-enter the download characters, since they were already sent to the printer with the first program. They will stay with the printer until you download new characters to replace them or turn the printer off. Even the $\langle \text{ESC} \rangle$ "@" command, which initializes the printer, does not destroy the contents of download RAM.

■ How to print the characters in control code area

As explained earlier, you can define characters with ASCII values below 32. But you cannot print out those characters with normal usage. This is inconvenient to us, even we can define download characters in that range. So, we have made a command to print those characters:

$\langle \text{ESC} \rangle$ "I" CHR\$(n)

This command is used to print the characters (if $n = 49$) or to select as the control codes (if $n = 48$). Let's try it out. Enter this program:

```

10 LPRINT CHR$(27);"x0";
20 LPRINT CHR$(27);"&";CHR$(0);CHR$(7);CHR$(7);
30 FOR M=1 TO 30
40 READ MM
50 LPRINT CHR$(MM);
60 NEXT M
70 LPRINT
90 DATA 1, 9, 2, 31,253, 64, 32, 1,128, 80,
      5, 0
100 DATA 168, 2,128, 80,133, 0,168, 74, 0, 80,
      52, 0
110 DATA 32, 8, 0, 0, 0, 0
120 LPRINT CHR$(27);"%1";CHR$(0);
130 LPRINT CHR$(27);"I1";
140 LPRINT CHR$(7)
150 LPRINT CHR$(27);"%0";CHR$(0)
160 LPRINT CHR$(27);"I0";

```

You should have printed out the character we defined, instead of beep.

THE PRINT MODE AND THE DOWNLOAD CHARACTERS

In the example program above, before we defined the new characters, we selected the Draft print mode. By doing this, we caused user-defined characters to print as draft characters because user-defined characters assume the mode that is in effect at the time they are defined.

The modes that affect user-defined characters are draft, and letter quality printing. (The $\langle \text{ESC} \rangle$ "x" n command selects between draft ($n = 0$) and letter quality ($n = 1$).

The commands for defining characters, selecting the download character set, and copying ROM are exactly the same for each of the user-defined print styles. The print mode in effect when the characters are defined determines the mode of the user-defined characters.

The print mode determines the size of the grid upon which the characters are designed and the speed at which they print. Table 8-1 shows the different character grid sizes for each of the print modes.

Table 8-1
Download character design grids

	$m1$ (Maximum)	$m0 + m1 + m2$ (Maximum)
Draft characters	9	12
Letter Quality characters	15	18

■ Draft download characters

As you've seen, draft characters are designed on a grid which is 9 dots wide by 24 dots high (plus up to three columns of dots for space between characters). Draft download characters print at the same high speed as normal draft characters, even though they may print with more dots and may even be designed for proportional spacing.

■ Letter quality download characters

If you select letter quality printing with the $\langle \text{ESC} \rangle$ "x" 1 command or by the control panel, you can design your own characters on a grid which is 15 dots wide by 24 dots high. Each character can be as wide as 18 dots, including space on either side of the character. The dot columns are spaced closer

together horizontally than draft style dot columns (the horizontal dot spacing is 1/180-inch when printing LQ characters — as opposed to 1/120-inch for draft characters).

Unlike draft characters, there are no restrictions on which dots can print. This, coupled with the closer dot spacing, allows you to design characters with higher resolution. The drawback is speed. Normal letter quality characters print more slowly than draft characters; the same is true of LQ download characters.

MEMO

CHAPTER 9

DOT GRAPHICS

Subjects we'll cover in Chapter 9 include—

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

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

COMPARING DOT GRAPHICS WITH DOWNLOAD CHARACTERS

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

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

So when do you use graphics and when do you use download characters? Practically anything you can do with graphics you can do with download characters, and vice versa. A clever programmer could actually plot a mathematical curve using

download characters or use strings of graphics data as user-defined characters. But why do it the hard way? There are several instances when dot graphics is clearly the best way to approach the problem:

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

USING THE DOT GRAPHICS COMMAND

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

⟨ESC⟩ “*” *n0 n1 n2 m1 m2*

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

Table 9-1
Graphics modes

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

■ Specifying the number of columns of dots

To figure the values of *n1* and *n2*, you’ll need to figure out how wide your graphics image will be. Then comes the fun part: converting one number (the number of columns of dots) into two! Why is it necessary to use two numbers to tell the printer

the number of graphics codes to expect? Because the largest number we can send in one byte (that's what the BASIC CHR\$() function sends: one byte) is 255. And with the normal density graphics it's possible to have a graphics image as wide as 816 dots. So to figure out how many columns of graphics data to expect, your printer multiplies $n2$ by 256 and adds the value of $n1$ to the product. If you divide the number of columns by 256, then $n2$ is the quotient and $n1$ is the remainder (why not let your computer figure it out for you: if the number of columns is assigned to variable X, then $n1 = X \text{ MOD } 256$ and $n2 = \text{INT}(X/256)$). Table 9-2 might make things even easier.

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

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

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

■ Specifying the graphics data

Now that we've told the printer how much data to expect, we better figure out how to send that information! Just as you do with download characters, with dot graphics you have control over firing of every single pin of the print head. When this printer produces 8-pin dot graphics modes, it prints with every third pin. It acts like a printer with nine wires that can only pro-

duce 8-pin graphics. In Figure 9-1, you can see that we've labelled each pin on the print head with a number as we did with download characters. And specifying pins to fire is done in the same way: to fire the second pin from the top, for instance, send a `CHR$(64)`. Firing several pins at once is done in a similar fashion. For example, to print the first, third, and fourth dots, add their values (128 + 32 + 16) to send this total: `CHR$(176)`. This is one byte of graphics data; it would replace `m1` in our format statement.

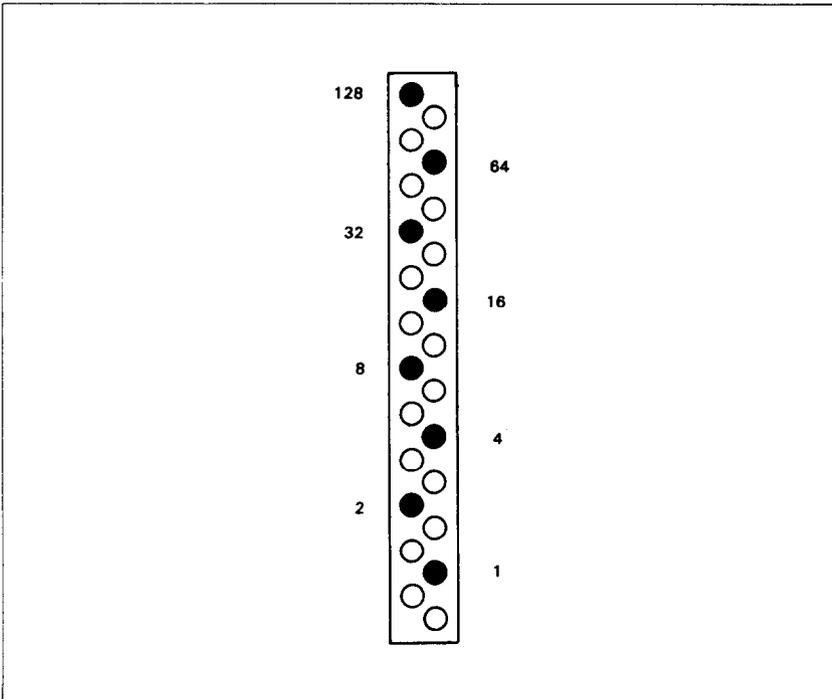


Figure 9-1. Starting with the most significant bit at the top, each third pin of the print head is assigned a value which is a power of two in case of 8-pin graphics modes.

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

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

```

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

```

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

■ Combining text and graphics

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

```

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

```

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

```

WOW!~~~~~THIS IS GREAT!

```

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

30 WID=1000

WOW!
THIS IS GREAT!



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

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

PRINTING A DESIGN OR LOGO

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

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

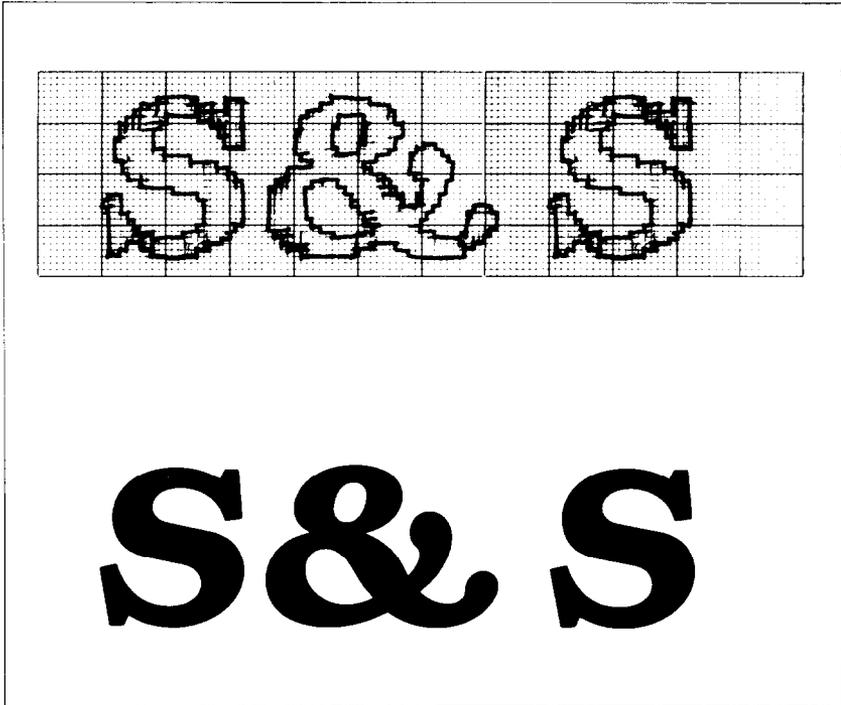


Figure 9-2. By laying out the logo on graph paper, you can calculate all of the graphics data.

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

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

```

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

```

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

```

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

```

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 60 dots per inch horizontally and 60 dots per inch vertically, it takes at least 480 bytes of memory for each square inch of plotted area. That doesn't sound so bad — but an area 12 inches square requires over 64K!)

Your array should be two-dimensional (just like graph paper) where one dimension will be the number of columns of dots and

the other dimension is the number of printing lines (remember that you can have up to eight rows of dots per printed line with the 8-pin graphics mode).

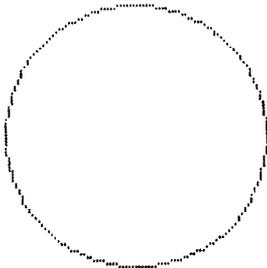
Here's a program that will use calculated-shape graphics to plot a circle. As you'll see, by changing a few lines it can be used to plot virtually any shape.

```

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

```

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



■ How the program works

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

distort it (you can, for example, make our circle print as an ellipse). Experiment a little bit!

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

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

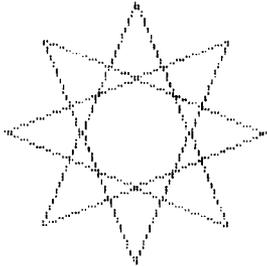
After these coordinates have been determined, the subroutine at line 400 is called. This routine turns “on” an individual dot in our array called BIT%. (Keep in mind that no printing has been done yet; the computer is still drawing the image on its “graph paper” in memory.) The way an individual dot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

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

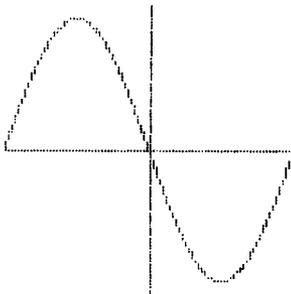
As you can see, by taking the program in small pieces and analyzing it, programming does not have to be difficult. If you want to try some other plots, try these (replace lines after 600 with the lines below). The printouts from each program are shown below the listing.

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

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



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



USING THE 24-PIN GRAPHICS MODE

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

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

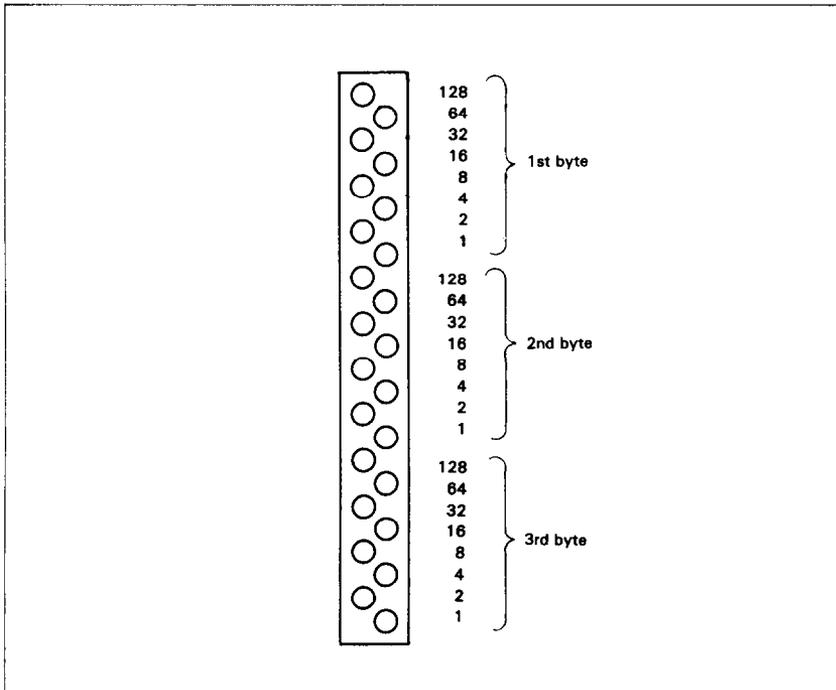


Figure 9-3. With the 24-pin graphics modes, each pin column of graphics is divided into three bytes.

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

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

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

```

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

```

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

The printout from this program is shown below.



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

```

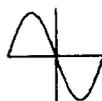
40 DIM BIT%(MAXCOL%/3,MAXROW%)
160 FOR ROW%=0 TO MAXROW%-1 STEP 3
170 PRINT#1, CHR$(27);"*";CHR$(39);CHR$(MAXCOL%);CHR$(0);
190 PRINT#1, CHR$(BIT%(COL%,ROW%+BYTE%));

```

```

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

```



COMPATIBILITY WITH EXISTING SOFTWARE

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

Table 9-3
Alternative graphics commands

Density	Single command	Individual command
Noraml	$\langle \text{ESC} \rangle$ "*" CHR\$(0) <i>n1 n2</i> <i>m1 m2 ...</i>	$\langle \text{ESC} \rangle$ "K" <i>n1 n2 m1</i> <i>m2 ...</i>
Double	$\langle \text{ESC} \rangle$ "*" CHR\$(1) <i>n1 n2</i> <i>m1 m2 ...</i>	$\langle \text{ESC} \rangle$ "L" <i>n1 n2 m1</i> <i>m2 ...</i>
Double	$\langle \text{ESC} \rangle$ "*" CHR\$(2) <i>n1 n2</i> <i>m1 m2 ...</i>	$\langle \text{ESC} \rangle$ "Y" <i>n1 n2 m1</i> <i>m2 ...</i>
Quadruple	$\langle \text{ESC} \rangle$ "*" CHR\$(3) <i>n1 n2</i> <i>m1 m2 ...</i>	$\langle \text{ESC} \rangle$ "Z" <i>n1 n2 m1</i> <i>m2 ...</i>

REDEFINING ALTERNATE GRAPHICS CODES

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

$\langle \text{ESC} \rangle$ "?" *n0 n1*

Where *n0* is one of the four letters, "K", "L", "Y", or "Z", and *n1* is one of the numbers used with the $\langle \text{ESC} \rangle$ "*" command, 0 to 4, 6, 32, 33, 38, and 39.

This is a quick way to change the aspect ratio of the design that you are printing. Changing the graphics mode will change

the width without changing the height. However, you should make this change with caution.

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

MEMO

CHAPTER 10

CARING FOR YOUR PRINTER

Subjects we'll cover in Chapter 10 include—

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

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

CLEANING THE PRINTER

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

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

REPLACING THE RIBBON

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

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

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

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

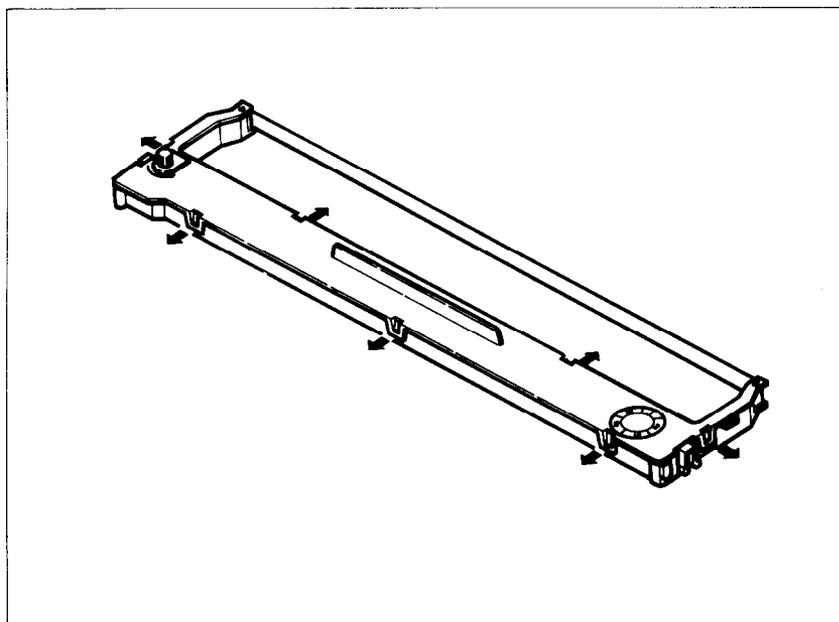


Figure 10-1. Unhook tabs to pry open the cartridge.

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

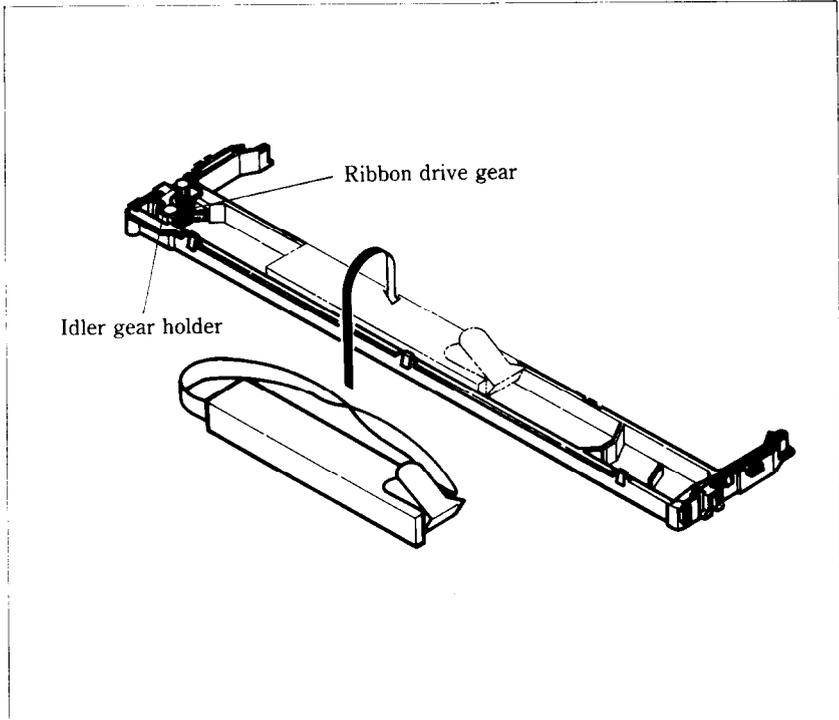


Figure 10-2. Replace the ribbon sub-cassette.

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

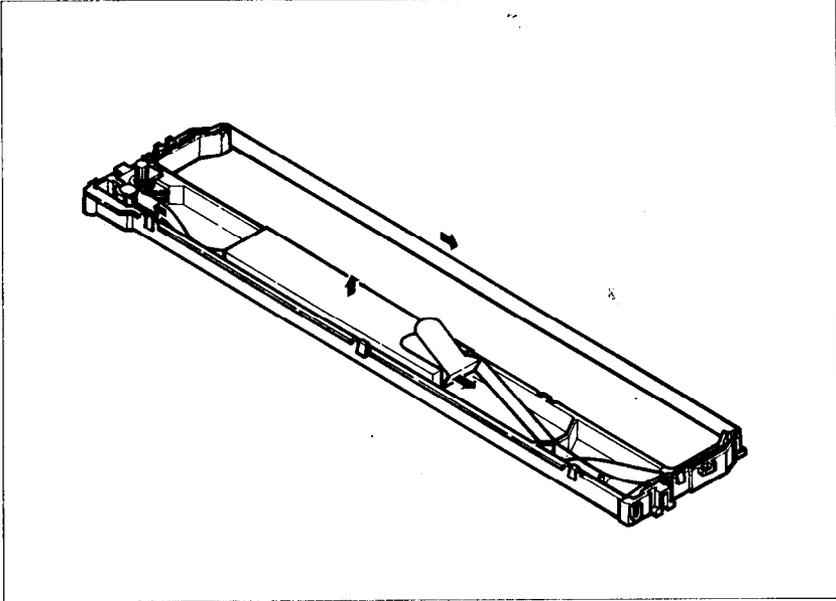


Figure 10-3. Make sure that the ribbon is not twisted when you thread it through its path.

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

REPLACING THE PRINT HEAD

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

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

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

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

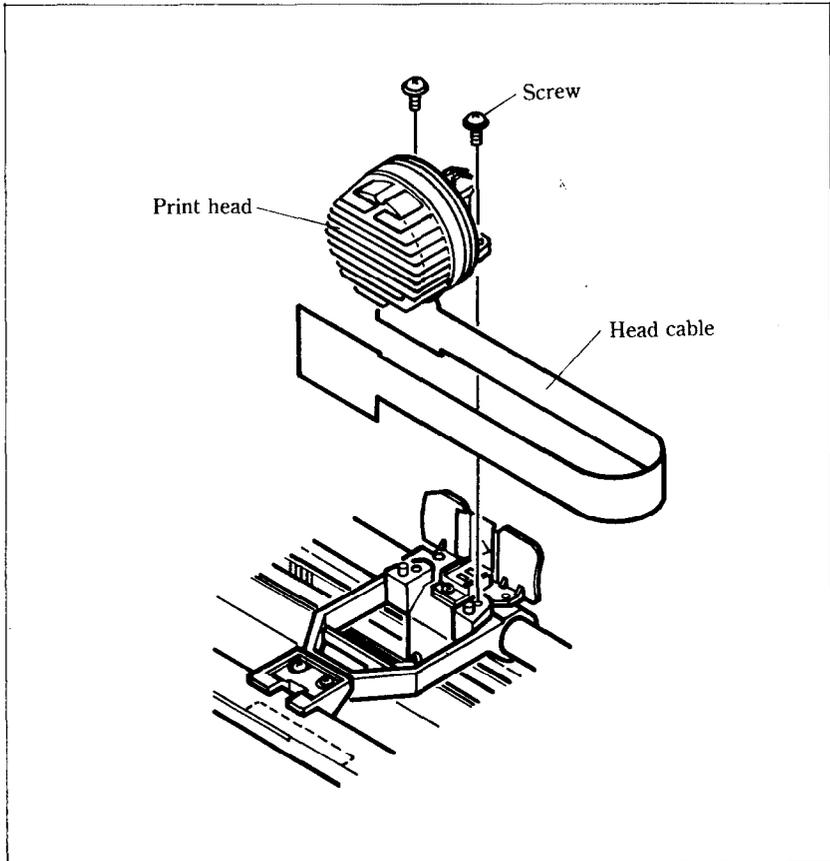


Figure 10-4. Replacement of the print head is simple.

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

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

MEMO

APPENDIX A

DIP SWITCH SETTINGS

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

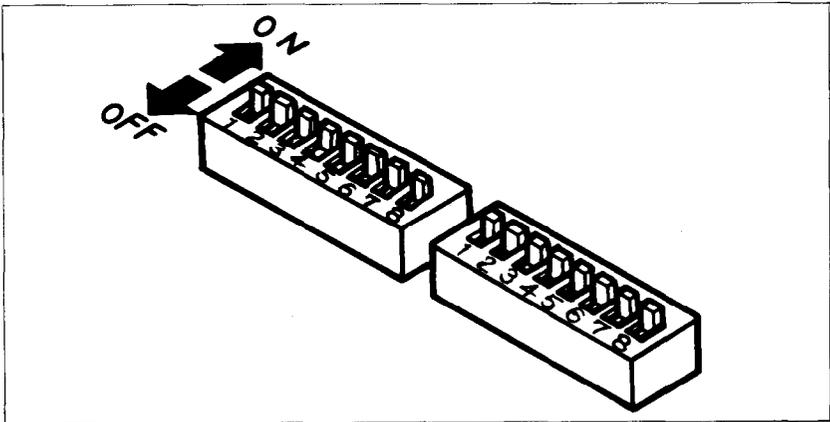


Figure A-1. A DIP switch is actually a series of several small switches.

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

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

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

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

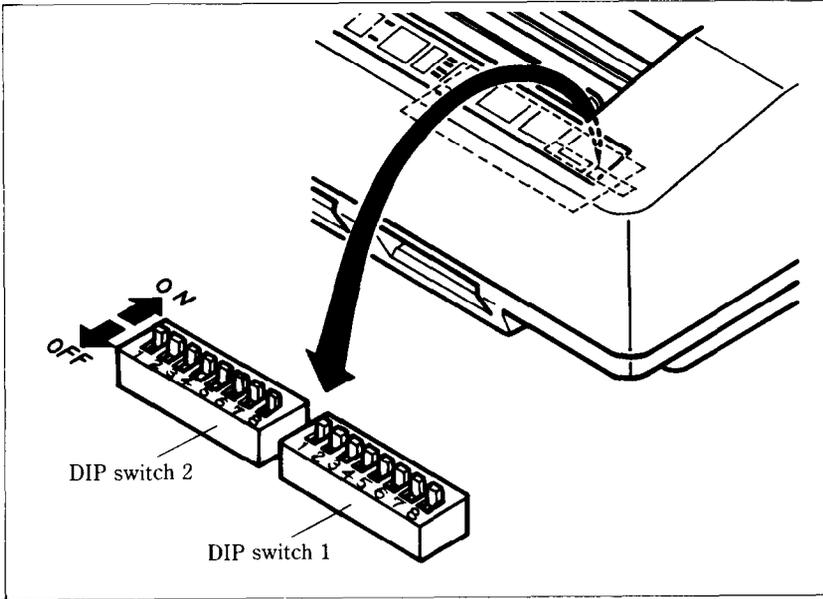


Figure A-2 The DIP switches are located under the printer cover.

Table A-1
DIP switch settings

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

SWITCH FUNCTIONS

Switch Function

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

Table A-2
International character sets

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

- 2-1 This switch determines the default bottom margin. When this switch is on, the bottom margin is not set at the power-on. When this switch is off, the bottom margin is automatically set to 1 inch. This switch is set on at the factory.
- 2-2 This switch selects the active control codes. Turn this switch on to use the "IBM" compatible mode. Turn this switch off to use the "Standard" mode. This switch is set on at the factory.
- 2-3 This switch controls the RAM condition. When this switch is on, the download character definitions are ignored and the RAM is used as the print buffer. When this switch is off, the download character definitions are enable to use and the print buffer is set to one line buffer. This switch is set on at the factory.
- 2-4 This switch disables the paper-out detector. If the switch is on, the printer will signal the computer when it runs out of paper and will stop printing. If the switch is off, the printer will ignore the paper-out detector and will continue printing. This switch is set on at the factory.
- 2-5 This switch controls the eighth bit of the parallel interface. If this switch is on, the printer will read all eight bits on the parallel interface. If this switch is off, the printer will only read the first seven bits on the parallel interface and ignores the eighth bit. This switch is set on at the factory.
- 2-6 When this switch is on, the computer must send a line feed command every time the paper is to advance. When this switch is off, the printer will automatically advance the paper one line every time it receives a carriage return. (Most BASICs send a line feed with every carriage return, therefore, this switch should usually be on.) This switch is set on at the factory.
- 2-7 This switch selects the print style of "zeroes". If this switch is on, the normal "zeroes" are printed. If this switch is off, the "slash zeroes" are printed. This switch is set on at the factory.

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

MEMO

APPENDIX B

ASCII CODES AND CONVERSION CHART

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

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

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

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

Standard ASCII Codes			Character set			
Decimal	Hexadecimal	Binary	Set1		Set2	
165	A5	1010 0101	N	N	N	N
166	A6	1010 0110	a	a	a	a
167	A7	1010 0111	o	o	o	o
168	A8	1010 1000	z	z	z	z
169	A9	1010 1001	l	l	l	l
170	AA	1010 1010	l	l	l	l
171	AB	1010 1011	½	½	½	½
172	AC	1010 1100	¼	¼	¼	¼
173	AD	1010 1101	i	i	i	i
174	AE	1010 1110	«	«	«	«
175	AF	1010 1111	»	»	»	»
176	B0	1011 0000				
177	B1	1011 0001				
178	B2	1011 0010				
179	B3	1011 0011				
180	B4	1011 0100	┆	┆	┆	┆
181	B5	1011 0101	┆	┆	┆	┆
182	B6	1011 0110				
183	B7	1011 0111				
184	B8	1011 1000	┆	┆	┆	┆
185	B9	1011 1001	┆	┆	┆	┆
186	BA	1011 1010				
187	BB	1011 1011	┆	┆	┆	┆
188	BC	1011 1100	┆	┆	┆	┆
189	BD	1011 1101	┆	┆	┆	┆
190	BE	1011 1110	┆	┆	┆	┆
191	BF	1011 1111	┆	┆	┆	┆
192	C0	1100 0000	┆	┆	┆	┆
193	C1	1100 0001	┆	┆	┆	┆
194	C2	1100 0010	T	T	T	T
195	C3	1100 0011	T	T	T	T
196	C4	1100 0100	—	—	—	—
197	C5	1100 0101	┆	┆	┆	┆
198	C6	1100 0110	┆	┆	┆	┆
199	C7	1100 0111	┆	┆	┆	┆
200	C8	1100 1000	┆	┆	┆	┆
201	C9	1100 1001	┆	┆	┆	┆
202	CA	1100 1010	┆	┆	┆	┆

Decimal	Standard ASCII Codes		Character set			
	Hexadecimal	Binary	Set1		Set2	
203	C B	1100 1011	T	T	T	T
204	C C	1100 1100	F	F	F	F
205	C D	1100 1101	=	=	=	=
206	C E	1100 1110	†	†	†	†
207	C F	1100 1111	±	±	±	±
208	D 0	1101 0000	⊥	⊥	⊥	⊥
209	D 1	1101 0001	T	T	T	T
210	D 2	1101 0010	T	T	T	T
211	D 3	1101 0011	⌞	⌞	⌞	⌞
212	D 4	1101 0100	⌞	⌞	⌞	⌞
213	D 5	1101 0101	F	F	F	F
214	D 6	1101 0110	F	F	F	F
215	D 7	1101 0111	†	†	†	†
216	D 8	1101 1000	†	†	†	†
217	D 9	1101 1001	⌋	⌋	⌋	⌋
218	D A	1101 1010	⌋	⌋	⌋	⌋
219	D B	1101 1011	■	■	■	■
220	D C	1101 1100	■	■	■	■
221	D D	1101 1101	■	■	■	■
222	D E	1101 1110	■	■	■	■
223	D F	1101 1111	■	■	■	■
224	E 0	1110 0000	α	α	α	α
225	E 1	1110 0001	β	β	β	β
226	E 2	1110 0010	Γ	Γ	Γ	Γ
227	E 3	1110 0011	τ	π	τ	π
228	E 4	1110 0100	Σ	Σ	Σ	Σ
229	E 5	1110 0101	σ	σ	σ	σ
230	E 6	1110 0110	μ	μ	μ	μ
231	E 7	1110 0111	τ	τ	τ	τ
232	E 8	1110 1000	φ	φ	φ	φ
233	E 9	1110 1001	θ	θ	θ	θ
234	E A	1110 1010	Ω	Ω	Ω	Ω
235	E B	1110 1011	δ	δ	δ	δ
236	E C	1110 1100	∞	∞	∞	∞
237	E D	1110 1101	φ	φ	φ	φ
238	E E	1110 1110	ε	ε	ε	ε
239	E F	1110 1111	∩	∩	∩	∩
240	F 0	1111 0000	≡	≡	≡	≡
241	F 1	1111 0001	±	±	±	±

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

MEMO

APPENDIX C

FUNCTION CODES

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

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

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

COMMANDS TO CONTROL PRINT STYLE

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

■ Font style controls

PURPOSE **Selects the LQ characters.**

CODE	<ESC>	“x”	1
(decimal ASCII)	27	120	1
(hex ASCII)	1B	78	01

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

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

SEE Chapter 5

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

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

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

PURPOSE **Selects an international character set.**

CODE	<ESC>	"R"	<i>n</i>
(decimal ASCII)	27	82	<i>n</i>
(hex ASCII)	1B	52	<i>n</i>

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

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

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

SEE Chapter 7

PURPOSE **Selects a character set.**

CODE	<ESC>	"k"	<i>n</i>
(decimal ASCII)	27	107	<i>n</i>
(hex ASCII)	1B	6B	<i>n</i>

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

SEE Chapter 5

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

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

■ Font pitch controls

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

PURPOSE	Sets the print pitch to elite.	
CODE	⟨ESC⟩	“M”
(decimal ASCII)	27	77
(hex ASCII)	1B	4D
REMARKS	This command causes printing to be done in elite pitch with 163 characters per line. This command is ignored when the “Panel” mode is selected at the power-on.	
SEE	Chapter 5	
PURPOSE	Sets the printer to condensed print.	
CODE	⟨SI⟩	
(decimal ASCII)	15	
(hex ASCII)	0F	
REMARKS	This command causes printing to be done in condensed pitch with 222 characters per line for pica condensed, and 234 characters per line for elite condensed (LQ characters are not printed in condensed pitch). You can select the pica condensed pitch with the control panel, but you cannot select the elite condensed pitch manually. This command is ignored when the “Panel” mode is selected at the power-on.	
SEE	Chapter 5	
PURPOSE	Sets the printer to condensed print.	
CODE	⟨ESC⟩	⟨SI⟩
(decimal ASCII)	27	15
(hex ASCII)	1B	0F
REMARKS	Same as ⟨SI⟩, above.	
SEE	Chapter 5	

PURPOSE	 Cancels the condensed print.		
CODE	〈DC2〉		
(decimal ASCII)	18		
(hex ASCII)	12		
REMARKS	This command cancels the condensed printing and returns the printer to the normal print pitch. This command is ignored when the "Panel" mode is selected at the power-on.		
SEE	Chapter 5		
PURPOSE	 Sets the printer to expanded print.		
CODE	〈ESC〉	"W"	1
(decimal ASCII)	27	87	1
(hex ASCII)	1B	57	01
REMARKS	This command causes characters to be printed twice as wide as normally (half the current pitch) until expanded printing is cancelled. Note: The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.		
SEE	Chapter 5		
PURPOSE	 Cancels the expanded print.		
CODE	〈ESC〉	"W"	0
(decimal ASCII)	27	87	0
(hex ASCII)	1B	57	00
REMARKS	This command resets the character pitch to what it was before expanded printing was set. Note: The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.		
SEE	Chapter 5		

PURPOSE **Sets the printer to expanded print for the remainder of the current line.**

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

REMARKS This command causes characters to be printed twice as wide as normally until a carriage return is sent. It also cancelled with <DC4>.

SEE Chapter 5

PURPOSE **Sets the printer to expanded print for the remainder of the current line.**

CODE **<ESC>** **<SO>**
(decimal ASCII) **27** **14**
(hex ASCII) **1B** **0E**

REMARKS Same as <SO>, above.

SEE Chapter 5

PURPOSE **Cancels one line expanded print.**

CODE **<DC4>**
(decimal ASCII) **20**
(hex ASCII) **14**

REMARKS This command cancels one line expanded print set with <SO> or <ESC> <SO>.

SEE Chapter 5

PURPOSE Sets the printer to proportional print.

CODE	<ESC>	"p"	1
(decimal ASCII)	27	112	1
(hex ASCII)	1B	70	01

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

SEE Chapter 5

PURPOSE Cancels proportional print.

CODE	<ESC>	"p"	0
(decimal ASCII)	27	112	0
(hex ASCII)	1B	70	00

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

SEE Chapter 5

■ Special print modes

PURPOSE **Sets the master print mode.**

CODE	⟨ESC⟩	“!”	<i>n</i>
(decimal ASCII)	27	33	<i>n</i>
(hex ASCII)	1B	21	<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, boldface, underlining, and any combination of these as determined by *n*, a number from 0 to 255. (See Table 5-10 for details.)

SEE Chapter 5

PURPOSE **Selects boldface printing.**

CODE	⟨ESC⟩	“E”
(decimal ASCII)	27	69
(hex ASCII)	1B	45

REMARKS This command causes characters to be printed in boldface until boldface is cancelled. You cannot be used with superscripts, or subscripts in boldface printing.

SEE Chapter 5

PURPOSE **Selects boldface printing.**

CODE	⟨ESC⟩	“G”
(decimal ASCII)	27	71
(hex ASCII)	1B	47

REMARKS Same as ⟨ESC⟩ “E”, above.

SEE Chapter 5

PURPOSE **Cancels boldface printing.**

CODE	<ESC>	"F"
(decimal ASCII)	27	70
(hex ASCII)	1B	46

REMARKS This command cancels boldface printing and returns the printer to normal printing.

SEE Chapter 5

PURPOSE **Cancels boldface printing.**

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

REMARKS Same as <ESC> "F", above.

SEE Chapter 5

PURPOSE **Selects underlining.**

CODE	<ESC>	"_"	1
(decimal ASCII)	27	45	1
(hex ASCII)	1B	2D	01

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

SEE Chapter 5

PURPOSE **Cancels underlining.**

CODE	<ESC>	"_"	0
(decimal ASCII)	27	45	0
(hex ASCII)	1B	2D	00

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

SEE Chapter 5

PURPOSE**Selects overlining.**

CODE	<ESC>	"_"	1
(decimal ASCII)	27	95	1
(hex ASCII)	1B	5F	01

REMARKS

This command overlines the following characters until overlining is cancelled.

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

SEE

Chapter 5

PURPOSE **Cancels overlining.**

CODE	<ESC>	"_"	0
(decimal ASCII)	27	95	0
(hex ASCII)	1B	5F	00

REMARKS

This command stops overlining.

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

SEE

Chapter 5

PURPOSE**Selects superscripts.**

CODE	<ESC>	"S"	0
(decimal ASCII)	27	83	0
(hex ASCII)	1B	53	00

REMARKS

This command raises the following characters and prints them as superscripts until superscripting is cancelled. Superscripts are not printed as elite pitch or condensed pitch. They are always printed with pica or expanded pica pitch.

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

SEE

Chapter 5

PURPOSE**Selects subscripts.**

CODE	<ESC>	"S"	1
(decimal ASCII)	27	83	1
(hex ASCII)	1B	53	01

REMARKS

This command lowers the following characters and prints as subscripts until subscripting is cancelled. All conditions described for superscripts also apply to subscripts.

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

SEE

Chapter 5

PURPOSE**Cancels a superscript or subscript.**

CODE	<ESC>	"T"
(decimal ASCII)	27	84
(hex ASCII)	1B	54

REMARKS

This command stops printing of superscripts or subscripts and returns to the normal printing previously set.

SEE

Chapter 5

CONTROLLING THE VERTICAL PRINT POSITION

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

■ Line feed controls

PURPOSE **Advances the paper one line (line feed).**

CODE	⟨LF⟩
(decimal ASCII)	10
(hex ASCII)	0A

REMARKS The actual distance advanced by the line feed is set either through DIP switch 2-8 or through various codes which can be sent (see below). When the DIP switch 2-6 is off, a line feed is automatically generated whenever the printer receives a carriage return.

SEE Chapter 6

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

CODE	⟨ESC⟩	“0”
(decimal ASCII)	27	48
(hex ASCII)	1B	30

REMARKS This command sets the actual distance the paper advances during all subsequent line feeds to 1/8 inch. You can select 1/8 inch line spacing as the power-on default by turning DIP switch 2-8 off.

SEE Chapter 6

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

CODE	⟨ESC⟩	“1”	
(decimal ASCII)	27	49	
(hex ASCII)	1B	31	

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

SEE Chapter 6

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

CODE	⟨ESC⟩	“3”	n
(decimal ASCII)	27	51	n
(hex ASCII)	1B	33	n

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

SEE Chapter 6

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

CODE	⟨ESC⟩	“A”	n
(decimal ASCII)	27	65	n
(hex ASCII)	1B	41	n

REMARKS This command works in two different functions depending on the setting of DIP switch 2-2. When the DIP switch 2-2 is set on, this command defines the actual distance the paper advances during all subsequent line feeds to $n/60$ inch. This command must be used in conjunction with ⟨ESC⟩ “2” which activates the ⟨ESC⟩ “A” definition. When the DIP switch 2-2 is set off, this command sets the actual distance the paper advances during all subsequent line feeds to $n/60$ inch immediately. The value of n must be between 1 and 255.

SEE Chapter 6

PURPOSE **Sets line spacing to 1/6 inch, or Uses \langle ESC \rangle "A" definition.**

CODE	\langle ESC \rangle	"2"
(decimal ASCII)	27	50
(hex ASCII)	1B	32

REMARKS This command works in two different functions depending on the setting of DIP switch 2-2. When the DIP switch 2-2 is set on, this command activates the line spacing defined in the \langle ESC \rangle "A" command. If the \langle ESC \rangle "A" command has not been defined, the line spacing is changed to 1/6 inch. When the DIP switch 2-2 is set off, this command sets the actual distance the paper advances during all subsequent line feeds to 1/6 inch. You can select the 1/6 inch line spacing as the power-on default by turning DIP switch 2-8 off.

SEE Chapter 6

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

CODE	\langle ESC \rangle	"J"	n
(decimal ASCII)	27	74	n
(hex ASCII)	1B	4A	n

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

SEE Chapter 6

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

CODE	<ESC>	"j"	n
(decimal ASCII)	27	106	n
(hex ASCII)	1B	6A	n

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

SEE Chapter 6

PURPOSE Sets print position to n lines.

CODE	<ESC>	"f"	1	n
(decimal ASCII)	27	102	1	n
(hex ASCII)	1B	66	01	n

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

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

SEE Chapter 6

■ Form feed and related commands

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

CODE <FF>
 (decimal ASCII) 12
 (hex ASCII) 0C

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

SEE Chapter 6

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

CODE <ESC> "C" 0 n
 (decimal ASCII) 27 67 0 n
 (hex ASCII) 1B 43 00 n

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

SEE Chapter 6

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

CODE	⟨ESC⟩	“C”	n
(decimal ASCII)	27	67	n
(hex ASCII)	1B	43	n

REMARKS This command sets the length of all subsequent pages to n lines. The value of n must be between 1 and 255. This command is ignored when the optional automatic sheet feeder is mounted on the printer.

SEE Chapter 6

■ Bottom margin and vertical tabs

PURPOSE **Sets the bottom margin.**

CODE	⟨ESC⟩	“N”	n
(decimal ASCII)	27	78	n
(hex ASCII)	1B	4E	n

REMARKS This command sets the bottom margin to n lines. The printer will generate a form feed whenever there are n lines left on the page. This command is ignored when the optional automatic sheet feeder is mounted on the printer. The value of n must be between 1 and 127.

SEE Chapter 6

PURPOSE **Cancels the bottom margin.**

CODE	⟨ESC⟩	“O”
(decimal ASCII)	27	79
(hex ASCII)	1B	4F

REMARKS This command cancels the bottom margin set by ⟨ESC⟩ “N” command.

SEE Chapter 6

PURPOSE Advances paper to the next vertical tab position.

CODE <VT>
 (decimal ASCII) 11
 (hex ASCII) 0B

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

SEE Chapter 6

PURPOSE Sets vertical tab positions.

CODE	<ESC>	"B"	<i>n1 n2 n3 ...</i>	0
(decimal ASCII)	27	66	<i>n1 n2 n3 ...</i>	0
(hex ASCII)	1B	42	<i>n1 n2 n3 ...</i>	00

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

SEE Chapter 6

PURPOSE Selects vertical channel.

CODE	<ESC>	"I"	<i>n0</i>
(decimal ASCII)	27	47	<i>n0</i>
(hex ASCII)	1B	2F	<i>n0</i>

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

SEE Chapter 6

PURPOSE Sets vertical tab positions in a channel.

CODE	<ESC>	"b"	<i>n0 n1 n2 n3 ...</i>	0
(decimal ASCII)	27	98	<i>n0 n1 n2 n3 ...</i>	0
(hex ASCII)	1B	62	<i>n0 n1 n2 n3 ...</i>	00

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

SEE Chapter 6

PURPOSE Sets vertical tab positions every *n* lines.

CODE	<ESC>	"e"	1	<i>n</i>
(decimal ASCII)	27	101	1	<i>n</i>
(hex ASCII)	1B	65	01	<i>n</i>

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

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

SEE Chapter 6

CONTROLLING THE HORIZONTAL PRINT POSITION

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

PURPOSE Returns print head to the left margin (carriage return).

CODE <CR>
(decimal ASCII) 13
(hex ASCII) 0D

REMARKS This command returns the print head to the left margin. If DIP switch 2-6 has been set off, then this command will also cause a line feed character to be generated after the carriage return, thereby advancing to the beginning of the next print line automatically.

SEE Chapter 6

PURPOSE Sets the left margin.

CODE <ESC> "I" *n*
(decimal ASCII) 27 108 *n*
(hex ASCII) 1B 6C *n*

REMARKS This command sets the left margin to *n* characters. Each line will begin in the (*n* + 1)th character position from the left edge. The value of *n* must be between 0 and 255. You can set the left margin manually with the control panel.

Note: Changing the print pitch after the left margin has been set does not change the margin — it stays in exactly the same place on the page.

SEE Chapter 6

PURPOSE **Sets the right margin.**

CODE	⟨ESC⟩	“Q”	<i>n</i>
(decimal ASCII)	27	81	<i>n</i>
(hex ASCII)	1B	51	<i>n</i>

REMARKS This command sets the right margin to *n*, which is the last character position that can be printed in a line. After execution of this command, any attempt to print beyond print position *n* will cause the printer to automatically generate a carriage return and a line feed before printing the remainder of the line. The value of *n* must be between 1 and 255. You can set the right margin manually with the control panel.

Note: Changing the print pitch after the right margin has been set does not change the margin — it stays in exactly the same position on the page.

SEE Chapter 6

PURPOSE **Moves the print head to the next horizontal tab position.**

CODE	⟨HT⟩
(decimal ASCII)	9
(hex ASCII)	09

REMARKS This command causes the print head to advance to the next horizontal tab position. The horizontal tab positions are set at power-on to print positions 8, 16, 24, etc. (to the maximum print position).

SEE Chapter 6

PURPOSE	Sets horizontal tab positions.			
CODE	<ESC>	"D"	<i>n1 n2 n3 ...</i>	0
(decimal ASCII)	27	68	<i>n1 n2 n3 ...</i>	0
(hex ASCII)	1B	44	<i>n1 n2 n3 ...</i>	00
REMARKS	This command cancels all current horizontal tab positions and sets those defined at print positions <i>n1</i> , <i>n2</i> , <i>n3</i> , etc. The maximum number of horizontal tab positions allowed is 32. The ASCII 0 character is used as a command terminator. Each horizontal tab position must be specified in ascending order.			
SEE	Chapter 6			
PURPOSE	Sets horizontal tab positions every <i>n</i> characters.			
CODE	<ESC>	"e"	0	<i>n</i>
(decimal ASCII)	27	101	0	<i>n</i>
(hex ASCII)	1B	65	00	<i>n</i>
REMARKS	This command cancels all current horizontal tab positions and sets those every <i>n</i> characters. Note: The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.			
SEE	Chapter 6			
PURPOSE	Moves the print head to an absolute horizontal position.			
CODE	<ESC>	"\$"	<i>n1</i>	<i>n2</i>
(decimal ASCII)	27	36	<i>n1</i>	<i>n2</i>
(hex ASCII)	1B	24	<i>n1</i>	<i>n2</i>
REMARKS	This command causes the printer to move the print head to an absolute horizontal position. The position, in inches, is determined by the formula $(n1 + n2 \times 256)/60$. The maximum position is 13.6 inches.			
SEE	Chapter 6			

PURPOSE	Moves the print head to a specified horizontal position.			
CODE (decimal ASCII) (hex ASCII)	⟨ESC⟩ 27 1B	“\” 92 5C	<i>n1</i> <i>n1</i> <i>n1</i>	<i>n2</i> <i>n2</i> <i>n2</i>
REMARKS	<p>This command causes the printer to move the print head to a specified horizontal position. It can move the print head either left or right. The distance, in inches, is determined by the following formulas:</p> <p>Draft: $(n1 + n2 \times 256)/120$ Letter Quality: $(n1 + n2 \times 256)/180$ Proportional: $(n1 + n2 \times 256)/360$</p> <p>To move to the left, add 64 to the calculated value of <i>n2</i>. The maximum distance is 13.6 inches. The command will be ignored if you try to move to a position outside of the current margins.</p>			
SEE	Chapter 6			
PURPOSE	Adds <i>n</i> dot spaces between characters.			
CODE (decimal ASCII) (hex ASCII)	⟨ESC⟩ 27 1B	“ <i>space</i> ” 32 20	<i>n</i> <i>n</i> <i>n</i>	
REMARKS	<p>This command increases the space between characters by <i>n</i> dots when the DIP switch 2-2 is set off.</p>			
SEE	Chapter 7			

PURPOSE	Sets the print position to n characters.			
CODE	<ESC>	"f"	0	n
(decimal ASCII)	27	102	0	n
(hex ASCII)	1B	66	00	n
REMARKS	This command sets the next print position to n columns from the left margin. The value of n must be between 0 and 127. Note: The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.			
SEE	Chapter 6			

DOWNLOAD CHARACTER COMMANDS

PURPOSE Defines download characters into RAM.

CODE	<ESC> "&"	0	<i>n1 n2 m0 m1</i> <i>m2 d1 d2 ... dx</i>
(decimal ASCII)	27	38	0 <i>n1 n2 m0 m1</i> <i>m2 d1 d2 ... dx</i>
(hex ASCII)	1B	26	00 <i>n1 n2 m0 m1</i> <i>m2 d1 d2 ... dx</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 between 33 and 126. Following *n2* this printer expects character data bytes for each character to be defined. The first byte, *m0*, specifies the left hand space of the download character. The second byte, *m1*, specifies the character width. And the third byte, *m2*, specifies the right hand space of the character. The sum of *m0*, *m1* and *m2* should be less than 12 for the draft characters and 18 for the LQ characters. *d1* through *dx* determine which dots form the character.

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

SEE Chapter 8

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

CODE	<ESC>	“.”	0	0	0
(decimal ASCII)	27	58	0	0	0
(hex ASCII)	1B	3A	00	00	00

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

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

SEE Chapter 8

PURPOSE **Selects download character set.**

CODE	<ESC>	“%”	1	0
(decimal ASCII)	27	37	1	0
(hex ASCII)	1B	25	01	00

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

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

SEE Chapter 8

PURPOSE **Cancels download character set.**

CODE	<ESC>	“%”	0	0
(decimal ASCII)	27	37	0	0
(hex ASCII)	1B	25	00	00

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

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

SEE Chapter 8

DOT GRAPHICS COMMANDS

PURPOSE	Prints normal-density graphics.
CODE	⟨ESC⟩ “K” <i>n1 n2 m1 m2.....</i>
(decimal ASCII)	27 75 <i>n1 n2 m1 m2.....</i>
(hex ASCII)	1B 4B <i>n1 n2 m1 m2.....</i>
REMARKS	This command selects 60 dots-per-inch, column-scan, bit-image graphics mode. The values of <i>n1</i> and <i>n2</i> represent the number of graphics characters to be printed, where the total number of characters = <i>n2</i> times 256 + <i>n1</i> . The correct number of graphics data bytes (<i>m1</i> , <i>m2</i> , etc.) must follow <i>n2</i> . The ASCII value of these characters determine which pins are fired for each character.
SEE	Chapter 9
PURPOSE	Prints double-density graphics.
CODE	⟨ESC⟩ “L” <i>n1 n2 m1 m2.....</i>
(decimal ASCII)	27 76 <i>n1 n2 m1 m2.....</i>
(hex ASCII)	1B 4C <i>n1 n2 m1 m2.....</i>
REMARKS	This command selects 120 dots-per-inch, column-scan, bit-image graphics mode. The values of <i>n1</i> and <i>n2</i> are the same as in normal-density graphics. The correct number of graphics data bytes (<i>m1</i> , <i>m2</i> , etc.) must follow <i>n2</i> . The ASCII value of these characters determine which pins are fired for each character.
SEE	Chapter 9

PURPOSE Prints double-density graphics.

CODE	⟨ESC⟩	“Y”	<i>n1 n2 m1 m2.....</i>
(decimal ASCII)	27	89	<i>n1 n2 m1 m2.....</i>
(hex ASCII)	1B	59	<i>n1 n2 m1 m2.....</i>

REMARKS Same as ⟨ESC⟩ “L”, above.

SEE Chapter 9

PURPOSE Prints quadruple-density graphics.

CODE	⟨ESC⟩	“Z”	<i>n1 n2 m1 m2.....</i>
(decimal ASCII)	27	90	<i>n1 n2 m1 m2.....</i>
(hex ASCII)	1B	5A	<i>n1 n2 m1 m2.....</i>

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

SEE Chapter 9

PURPOSE**Selects graphics modes.**

CODE	<ESC>	"*"	<i>n0 n1 n2 m1 m2.....</i>
(decimal ASCII)	27	42	<i>n0 n1 n2 m1 m2.....</i>
(hex ASCII)	1B	2A	<i>n0 n1 n2 m1 m2.....</i>

REMARKS

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

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

SEE

Chapter 9

PURPOSE	Redefines the graphics mode.			
CODE	⟨ESC⟩	“?”	<i>n0</i>	<i>n1</i>
(decimal ASCII)	27	63	<i>n0</i>	<i>n1</i>
(hex ASCII)	1B	3F	<i>n0</i>	<i>n1</i>
REMARKS	This command redefines one of the 4 alternate graphics commands — ⟨ESC⟩ “K”, ⟨ESC⟩ “L”, ⟨ESC⟩ “Y”, or ⟨ESC⟩ “Z” — as one of the nine graphics density numbers with the ⟨ESC⟩ “*” command, where <i>n0</i> is “K”, “L”, “Y”, or “Z” and <i>n1</i> is 0, 1, 2, 3, 4, 6, 32, 33, 38, or 39.			
SEE	Chapter 9			

OTHER COMMANDS

PURPOSE	Sets the value of the eighth data bit to logical 1.		
CODE	⟨ESC⟩	“)”	
(decimal ASCII)	27	62	
(hex ASCII)	1B	3E	
REMARKS	This command forces the eighth data bit of each subsequent character sent to the printer to logical 1. This code allows users with a 7-bit interface to access those characters whose ASCII code is greater than 127. This code should not be used to transmit printer control codes.		
SEE	Chapter 7		

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

CODE	⟨ESC⟩	“=”
(decimal ASCII)	27	61
(hex ASCII)	1B	3D

REMARKS This command forces the eighth data bit of each subsequent character sent to the printer to logical 0. This code should not be used to transmit printer control code.

SEE Chapter 7

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

CODE	⟨ESC⟩	“#”
(decimal ASCII)	27	35
(hex ASCII)	1B	23

REMARKS This command cancels either setting of the eighth data bit. The printer will use the value of the eighth data bit that is sent from the computer. This code allows users with a 7-bit interface to resume normal functions after accessing those characters whose ASCII code is greater than 127.

SEE Chapter 7

PURPOSE **Moves the print head back one print position (backspace).**

CODE	⟨BS⟩
(decimal ASCII)	8
(hex ASCII)	08

REMARKS This command shifts the print head one column to the left. If the print head is at the left margin, the command is ignored. This command can be used to overstrike characters.

SEE Chapter 7

PURPOSE	Deletes the last character sent.
CODE (decimal ASCII) (hex ASCII)	 127 7F
REMARKS	This command deletes the last character received. This command is ignored if the last character received has already been printed, or if the last character received was all or part of a function code.
SEE	Chapter 7
PURPOSE	 Cancels line.
CODE (decimal ASCII) (hex ASCII)	<CAN> 24 18
REMARKS	This command deletes the last line in the print buffer at the time the command is used.
SEE	Chaper 7
PURPOSE	Sets printer off line.
CODE (decimal ASCII) (hex ASCII)	<DC3> 19 13
REMARKS	This command causes the printer to set itself off line, disregarding all subsequent characters and function codes, with the exception of <DC1>, which will return the printer to an on line state. This is not the same as pushing the On Line key. When the On Line lamp is out the printer will not respond to <DC1>.
SEE	Chapter 7

PURPOSE	Sets printer on line.
CODE	⟨DC1⟩
(decimal ASCII)	17
(hex ASCII)	11
REMARKS	This command resets the printer to an on line state, thus allowing it to receive and process all subsequent characters and function codes. This is not the same as pushing the On Line key. When the On Line lamp is out the printer will not respond to ⟨DC1⟩.
SEE	Chapter 7
PURPOSE	Sounds the printer bell.
CODE	⟨BEL⟩
(decimal ASCII)	7
(hex ASCII)	07
REMARKS	This command causes the buzzer to sound for about a quarter of a second.
SEE	Chapter 7
PURPOSE	Disables paper-out detector.
CODE	⟨ESC⟩ "8"
(decimal ASCII)	27 56
(hex ASCII)	1B 38
REMARKS	This command causes the printer to disregard the signal sent by the paper-out detector. The paper-out signal normally sounds the printer bell and stops printing until paper is inserted and the printer is reset. DIP switch 2-4 can also set to disable the paper-out detector.
SEE	Chapter 7

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

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

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

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

CODE	<ESC>	“<”
(decimal ASCII)	27	60
(hex ASCII)	1B	3C

REMARKS This command immediately returns the print head to the left margin. The remainder of the line is printed from left to right. Normal (bi-directional) printing resumes following a carriage return.

SEE Chapter 7

PURPOSE **Selects double-height expanded printing.**

CODE	<ESC>	“h”	1
(decimal ASCII)	27	104	1
(hex ASCII)	1B	68	01

REMARKS This command causes the printer to print expanded characters with double-height.

SEE Chapter 7

PURPOSE **Selects quadruple-height expanded printing.**

CODE	<ESC>	“h”	2
(decimal ASCII)	27	104	2
(hex ASCII)	1B	68	02

REMARKS This command causes the printer to print expanded characters with quadruple-height.

SEE Chapter 7

PURPOSE Cancels double- and quadruple-height expanded printing.

CODE	<ESC>	"h"	0
(decimal ASCII)	27	104	0
(hex ASCII)	1B	68	00

REMARKS This command cancels double-height and quadruple-height expanded printing, and returns to the previous character size.

SEE Chapter 7

PURPOSE Prints characters in the undefined control code area.

CODE	<ESC>	"I"	1
(decimal ASCII)	27	73	1
(hex ASCII)	1B	49	01

REMARKS This command causes the printer to print characters in the undefined control code area. This command is ignored when the DIP switch 2-2 is set on.

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

SEE Chapter 8

PURPOSE Selects undefined codes as control codes.

CODE	<ESC>	"I"	0
(decimal ASCII)	27	73	0
(hex ASCII)	1B	49	00

REMARKS This command cancels to print the characters in the undefined control codes and restores them as the control codes.

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

SEE Chapter 8

PURPOSE	Sets immediate print mode.		
CODE	<ESC>	"i"	1
(decimal ASCII)	27	105	1
(hex ASCII)	1B	69	01
REMARKS	This command selects the immediate print mode. In the immediate print mode the print head prints one character at a time, as you send it. The printer also moves the paper up so that you can see the current line and then down to continue printing. This kind of instant feedback can be especially helpful in telecommunications.		
	Note: The character "1" (decimal code 49, hexadecimal code 31) can be used instead of ASCII 1.		

SEE Chapter 7

PURPOSE	 Cancels immediate print mode.		
CODE	<ESC>	"i"	0
(decimal ASCII)	27	105	0
(hex ASCII)	1B	69	00
REMARKS	This command cancels the immediate print mode and returns the normal print mode.		
	Note: The character "0" (decimal code 48, hexadecimal code 30) can be used instead of ASCII 0.		

SEE Chapter 7

PURPOSE	Repeats the characters.			
CODE	<ESC>	"V" <i>n m1 ... mx</i>	<ESC>	"V" 0
(decimal ASCII)	27	86 <i>n m1 ... mx</i>	27	86 0
(hex ASCII)	1B	56 <i>n m1 ... mx</i>	1B	56 00
REMARKS	This command repeats <i>n</i> times (<i>n</i> is an integer number from 0 to 255) data bytes represented by <i>m1 ... mx</i> .			

SEE Chapter 7

PURPOSE**Resets the printer.****CODE**

(decimal ASCII)

(hex ASCII)

⟨ESC⟩ “@”

27 64

1B 40

REMARKS

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

SEE

Chapter 7

PURPOSE Sets 9-pin graphics emulation mode.

CODE	<ESC>	"g"
(decimal ASCII)	27	103
(hex ASCII)	1B	67

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

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

<ESC>"A" *n* — Sets or defines line spacing to $n/72$ inch.

<ESC>"3" *n* — Sets line spacing to $n/216$ inch.

<ESC>"J" *n* — Sends a one-time paper feed of $n/216$ inch.

<ESC>"j" *n* — Sends a one-time reverse feed of $n/216$ inch.

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

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

SEE

Chapter 7

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

PURPOSE	Selects auto feed mode.				
CODE	"("	"("	"4"	"")	"")
(decimal ASCII)	40	40	52	41	41
(hex ASCII)	28	28	34	29	29
REMARKS	Same as <ESC> 4, above.				
SEE	Chapter 7				

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

PURPOSE	Cancels auto feed mode.				
CODE	"("	"("	"0"	"")	"")
(decimal ASCII)	40	40	48	41	41
(hex ASCII)	28	28	30	29	29
REMARKS	Same as <ESC> 0, above.				
SEE	Chapter 7				

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

PURPOSE	Supplies paper from first bin.				
CODE	"("	"("	"1"	")"	")"
(decimal ASCII)	40	40	49	41	41
(hex ASCII)	28	28	31	29	29
REMARKS	Same as <ESC> 1, above.				
SEE	Chapter 7				

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

PURPOSE	Supplies paper from second bin.				
CODE	"("	"("	"2"	")"	")"
(decimal ASCII)	40	40	50	41	41
(hex ASCII)	28	28	32	29	29
REMARKS	Same as <ESC> 2, above.				
SEE	Chapter 7				

PURPOSE**Ejects paper.****CODE**

(decimal ASCII)

(hex ASCII)

<ESC>		"R"
27	25	82
1B	19	52

REMARKS

This command causes the printer to eject paper. This command is ignored when the optional automatic sheet feeder is not mounted on the printer.

SEE

Chapter 7

PURPOSE**Ejects paper.****CODE**

(decimal ASCII)

(hex ASCII)

"("	"("	"R"	"")"	"")"
40	40	82	41	41
28	28	52	29	29

REMARKS

Same as **<ESC> "R"**, above.

SEE

Chapter 7

APPENDIX D

COMMAND SUMMARY IN NUMERIC ORDER

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

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

- <ESC> "B" $n_1 n_2 n_3 \dots$ CHR\$(0) Sets vertical tab positions
 <ESC> "C" CHR\$(0) n Sets page length to n inches
 <ESC> "C" n Sets page length to n lines
 <ESC> "D" $n_1 n_2 n_3 \dots$ CHR\$(0) Sets horizontal tab positions
 <ESC> "E" Selects boldface printing
 <ESC> "F" Cancels boldface printing
 <ESC> "G" Selects boldface printing
 <ESC> "H" Cancels boldface printing
 <ESC> "I" 0 Selects undefined codes as control codes
 <ESC> "I" 1 Prints characters in the undefined control code area
 <ESC> "J" n Sends a one-time paper feed of $n/180$ inch
 <ESC> "K" $n_1 n_2 m_1 m_2 \dots$ Prints normal-density graphics
 <ESC> "L" $n_1 n_2 m_1 m_2 \dots$ Prints double-density graphics
 <ESC> "M" Sets the print pitch to elite
 <ESC> "N" n Sets the bottom margin
 <ESC> "O" Cancels the bottom margin
 <ESC> "P" Sets the print pitch to pica
 <ESC> "Q" n Sets the right margin
 <ESC> "R" n Selects an international character set
 <ESC> "S" 0 Selects superscripts
 <ESC> "S" 1 Selects subscripts
 <ESC> "T" Cancels a superscript or subscript
 <ESC> "U" 0 Cancels uni-directional printing
 <ESC> "U" 1 Selects uni-directional printing
 <ESC> "V" $n m_1 \dots m_x$ <ESC>"V"CHR\$(0) Repeats the characters
 <ESC> "W" 0 Cancels the expanded print
 <ESC> "W" 1 Sets the printer to expanded print
 <ESC> "Y" $n_1 n_2 m_1 m_2 \dots$ Prints double-density graphics
 <ESC> "Z" $n_1 n_2 m_1 m_2 \dots$ Prints quadruple-density graphics
 <ESC> "\ " $n_1 n_2$ Moves the print head to a specified horizontal position
 <ESC> " _ " 0 Cancels overlining
 <ESC> " _ " 1 Selects overlining
 <ESC> "b" $n_0 n_1 n_2 n_3 \dots$ CHR\$(0) Sets vertical tab positions in a channel

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

APPENDIX E

TECHNICAL SPECIFICATIONS

Printing

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

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

Paper

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

Printer

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

Parallel interface

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

Serial interface (option)

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

MEMO

APPENDIX F

THE

PARALLEL INTERFACE

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

Data transfer rate:	1,000 to 6,000 characters per second
Synchronization:	Via externally supplied <u>STROBE</u> pulses
Handshaking:	ACK and BUSY signals
Logic level:	Compatible with TTL level

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

■ Functions of the Connector Signals

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

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

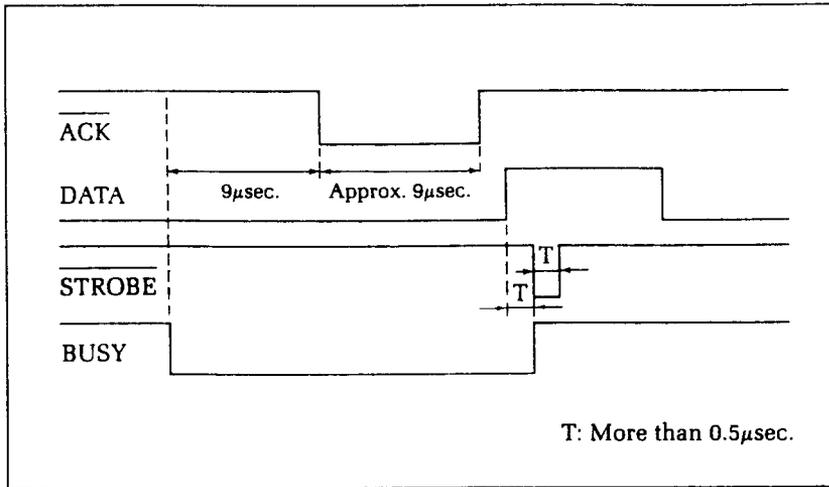


Figure F-1. The interface timing diagram.

Signal Name	Circuit Example
DATA 1 - DATA 8 (To Printer)	<p>74LS Compatible</p>
$\overline{\text{STROBE}}$ (To Printer)	<p>74LS Compatible</p>
BUSY, $\overline{\text{ACK}}$ (From Printer)	<p>74LS Compatible</p>

Figure F-2. Typical interface circuit.

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

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

The printer will report that it has run out of paper by making

the PAPER OUT signal on pin 12 high. This pin can be held low by turning DIP switch 2-4 off. When the printer is in the on-line state pin 13 is held high. This signal (SELECTED) tells the computer that the printer is ready to receive data.

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

Table F-1
Parallel interface pin functions

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

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

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

APPENDIX G

SERIAL INTERFACE

SPECIFICATIONS

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

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

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

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

CONFIGURING THE SERIAL INTERFACE

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

Table G-1
Serial interface pin functions

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

Table G-2
DIP switch on serial board

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

Table G-3
Handshaking protocols

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

Table G-4
Data transfer rates

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

THE SERIAL PROTOCOLS

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

■ Serial busy protocols

In the serial busy protocols, this printer uses DTR (pin 20) and RCH (pin 11) to signal to the computer when it is able to accept

data. These two pins go ON when the printer is ready to accept data. In the 1 byte mode they go OFF after each character is received. In the 1 block mode they only go OFF when the printer's buffer approaches capacity. In both cases they will stay OFF if the buffer is too full to accept more data.

■ XON/XOFF protocol

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

■ ACK protocol

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

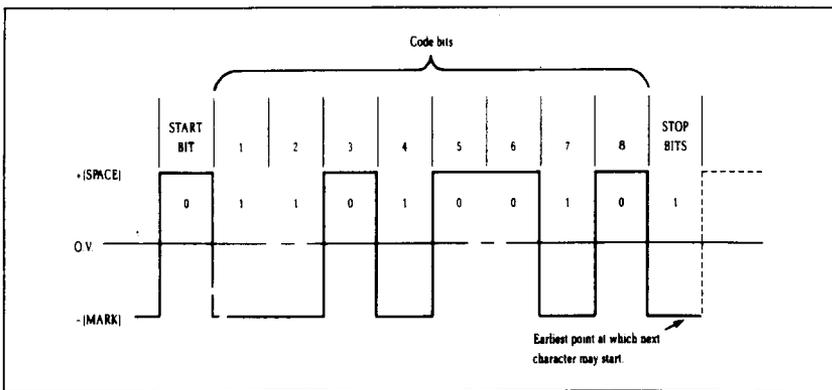


Figure G-1. Typical data byte on the serial interface.

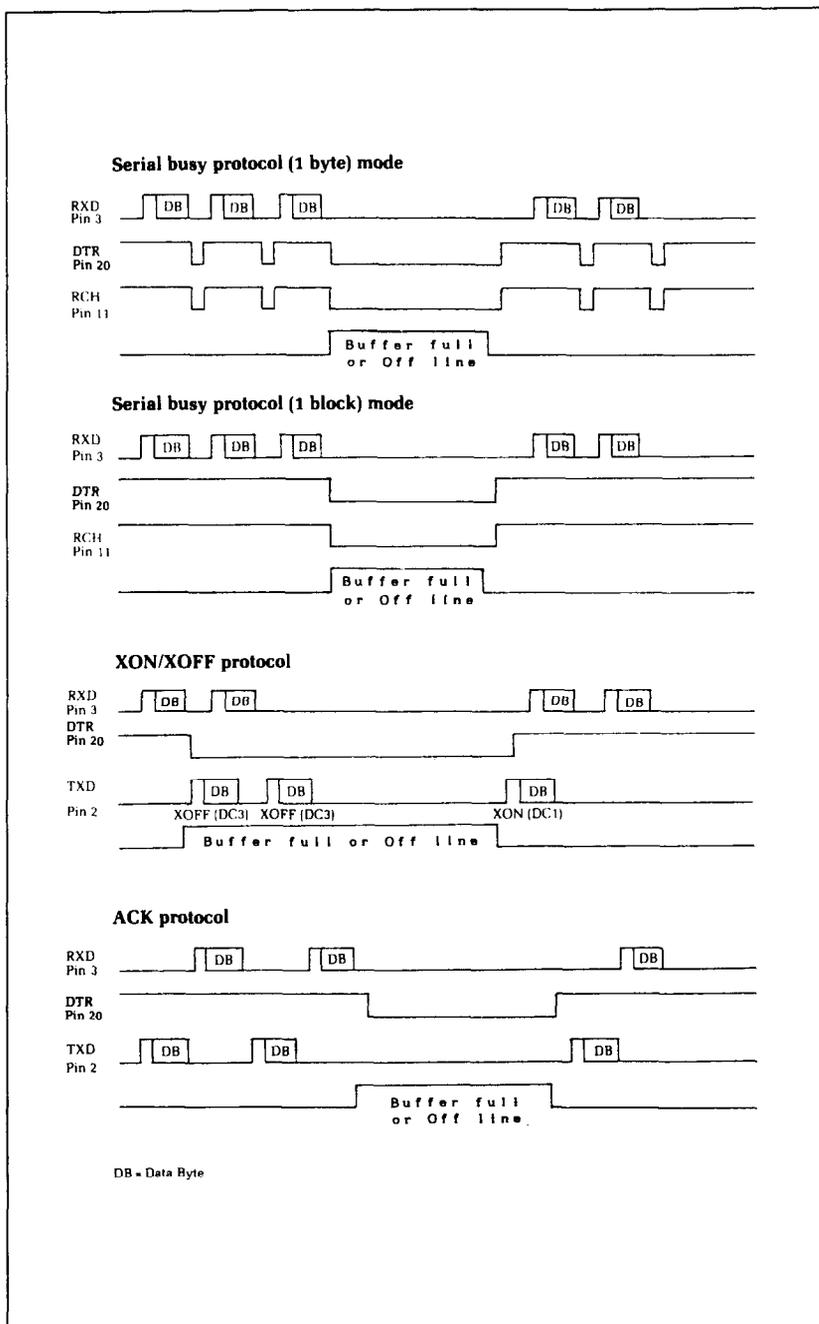


Figure G-2. Serial protocol timing charts.

MEMO

APPENDIX H

CONNECTING WITH COMPUTER

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

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

CONNECTING WITH IBM-PC AND COMPAQ

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

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

Table H-1
IBM-PC parallel cable

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

■ BASIC programming

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

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

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

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

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

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

■ Listing programs

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

CONNECTING WITH APPLE II COMPUTERS

Apple II computers require an interface board (mounted inside the Apple II) and a cable to run this printer. We recommend that you use the **grafstar**TM interface for the Apple II, II + ,

and IIe. It comes complete with a cable and is easily installed. A unique feature of the **grafstar**TM makes it possible to do some fancy dot graphics programming.

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

Table H-2
Apple parallel cable

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

■ Applesoft BASIC

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

```
10 PR#1
```

or

```
10 PRINT "<Ctrl-D> PR#1"
```

or

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

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

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

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

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

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

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

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

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

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

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

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

■ Listing programs

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

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

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

CONNECTING WITH TRS-80 COMPUTERS

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

Table H-3
TRS-80 Model I parallel cable

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

Table H-4
TRS-80 Model II parallel cable

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

■ TRS-80 BASIC

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

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

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

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

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

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

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

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

■ Listing programs

To list a BASIC program that is in your TRS-80's memory on

this printer, type LLIST. This directs the listing to the printer instead of the screen.

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

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

Table H-5
Kaypro parallel cable

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

Table H-6
Osborne 1 parallel cable

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

■ Using MBASIC

Many CP/M computers use Microsoft BASIC (called MBASIC). MBASIC is a very close relative of the IBM-Microsoft BASIC. The only difference is that MBASIC "inter-

prets" CHR\$(9) and substitutes a group of spaces to simulate a tab. You can send a horizontal tab to this printer by using CHR\$(137) instead of CHR\$(9).

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

```
10 WIDTH LPRINT 255
```

■ Listing programs

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